PL-TR-94-2271

USER'S GUIDE FOR THE TOPSIDE IONOSPHERIC PLASMA MONITOR (SSIES, SSIES-2 AND SSIES-3) ON THE SPACECRAFT OF THE DEFENSE METEOROLOGICAL SATELLITE PROGRAM (DMSP) VOLUME III: PROGRAM MAINTENANCE MANUAL

J. Robert Cornelius
Andrew J. Mazzella, Jr.

RDP Incorporated
391 Totten Pond Road
Waltham, Massachusetts 02154

11 October 1994

Scientific Report No. 4

Approved for public release; distribution unlimited

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AIR FORCE BASE, MA 01731-3010

"This technical report has been reviewed and is approved for publication"

Edward C. Robinson

EDWARD C. ROBINSON
Contract Manager
Data Analysis Division

Robert E. McInerney

ROBERT E. McINERNEY, Director
Data Analysis Division

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/IM, 29 Randolph Road, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	11 October 1994	Scientific Report No. 4

4. TITLE AND SUBTITLE

User's Guide for the Topside Ionospheric Plasma Monitor (SSIES, SSIES-2 and SSIES-3) on the Spacecraft of the Defense Meteorological Satellite Program (DMSP), Volume III: Program Maintenance Manual

5. FUNDING NUMBERS

PE 35160F
PR 7659 TA 05 WU AC

Contract: F19628-89-C-0079

6. AUTHOR(S)

J. Robert Cornelius
Andrew J. Mazzella, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

RDP Incorporated
391 Totten Pond Road
Waltham, Massachusetts 02154

8. PERFORMING ORGANIZATION REPORT NUMBER

RDP-TR-9402

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Phillips Laboratory
29 Randolph Road
Hanscom Air Force Base, Massachusetts 01731-3010
Contract Manager: Edward C. Robinson/GPD

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

PL-TR-94-2271

11. SUPPLEMENTARY NOTES

Supersedes PD-NW-85-336R, Volume III, June 1987, and NWRA-87-R011, Volume III, August 1987.

Volume I: PL-TR-94-2187
Volume II: PL-TR-94-2270

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of the Program Maintenance Manual for the DMSP SSIES flight data processing software is to provide AFSFC programming personnel with the information necessary to understand and maintain the various components of the system. This software has been specifically developed to process the SSIES-2 and SSIES-2A data formats, with provisions for retroactive adaptation to the original SSIES format and future adaptation to the SSIES-3 format.

This Program Maintenance Manual describes the three programs which constitute the AFSFC SSIES processing system: BNBA, LDCON02 and APGA. The BNBA program performs data format conversions for the various SSIES telemetry data formats to generate a common file format for subsequent processing by the APGA program. The LDCON02 program generates the reference parameter file of satellite and instrument conversion constants and processing options for use by the APGA program. The APGA program performs quality evaluations and analyses of the SSIES telemetry data to create database files and reports of quantities which characterize the ionosphere.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Ionosphere, Plasma Analysis, Particle Detectors	308
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

Table of Contents

Figures

Tables

SECTION 1. GENERAL

1.1 Purpose of the Program Maintenance Manual.

The objective of this Program Maintenance Manual for the programs which make up the SSIES Flight Data Processing (FDP) system is to provide the maintenance programmer with the information necessary to effectively maintain the system.

1.2 Project References.

1.2.1 System Nature.

The SSIES FDP system will process flight data from the DMSP Mission Sensors for Ions, Electrons, and Scintillation (SSIES) to provide geophysical parameters describing the earth's ionosphere. This system will be part of the overall AFSFC DMSP Mission Sensor system. The SSIES FDP system consists of three programs:

1. Program BNBA reads, unpacks, and checks the Ephemeris and raw flight data from IES, IES2, IES2A, or IES3 satellite configurations. The data are reformatted into the standard format and written to the IESPREPFILE for input to the APGA program.

2. Program APGA processes the standard format flight data to convert it to geophysical parameters which are stored in Environmental Data Records (EDRs) in an EDR file (IESEDRFILE). Summary records of the EDRs are written to files for transfer to the Astrogeophysical Data Base (AGDB).

3. Program LDCON allows the user to interface with the file which contains control parameters for program APGA.

1.2.2 Project Participants.

a. Sponsor: Air Force Space and Missile Command (USAF Materiel Command) Space Division, DMSP Program Office (SMC/CIE)

b. Developer: RDP Incorporated, based upon a system originally developed by Northwest Research Associates, Inc. (NWRA)

c. User: Air Force Space Forecast Center (AFSFC)

d. Operating Center: Air Force Space Forecast Center (AFSFC)

1.2.3 Project Documents

a. (Project Request) USAF Space Division Contract No. F04701-84-C-0038, Task 17-09.

b. AFGL-TR-78-0071, <u>The Topside Ionosphere Plasma Monitor (SSIE) for the Block 5D/Flight 2 DMSP Satellite</u>, March 1978, ADA058503.

c. AFGL-TR-80-0152, <u>In-Flight Characteristics of the Topside Ionosphere Monitor (SSIE) on the DMSP Satellite Flight 2 and Flight 4</u>, April 1980, ADA088879.

d. AFGL-TR-84-0103, Drift Scintillation Meter, March 1984, ADA142523.

e. Analytyx Report A30803R, SSIES Plasma Monitor System Main Electronics Package - R&D Equipment Information Report, prepared for AFGL contract No. F19628-80-C-0162, August 1983.

f. Development of Software for the Analysis of Plasma Measurements Using the Retarding Potential Analyzer, Unpublished, undated document provided by AFGL/PHG, June 1985. (copy in SSIES Project Workbook).

g. SSIES Digital Flight Data Processing Program Functional Description, Unpublished draft document provided by AFGL/PHG, July 1984 (copy in SSIES Project Workbook).

h. DOD Standard 7935, Automated Data Systems Documentation, 15 February 1983.

i. Sperry Fortran (ASCII) Level 10R1 Programmer Reference.

j. AFGWC Software Standards, 1 July 1985.

k. AFGWC Systems Documentation, Library Software (FLASH/AFLASH).

l. AFGWC Systems Documentation, Data Format Handbook.

m. AFGWC/SDMS, DMSP Block 5D2 Special Sensor Data Processing System Functional Description, Revision 2, 10 December 1984.

n. AFGWC/SDMS, DMSP Block 5D2 Special Sensor Data Processing System Data Base Specification, May 1985.

o. PD-NW-83-292R, Definition of Procedures for Gridding the WBMOD Ionosphere Scintillation Model, April 1983.

p. PD-NW-85-336R, DMSP SSIES Flight Data Processor System Documentation,
   1. Volume I, Functional Description
   2. Volume II, Users Manual
   3. Volume IV, Test Plan
   4. Volume V, Test Analysis Report.

q. PD-NW-82-273R, WBMOD Scintillation Model System Documentation, Volume III, Program Maintenance Manual, 29 May 1985.

r. SSIES Project Workbook, August 1986.

s. Livingston, R.C., C.L. Rino, J.P. McClure, and W.B. Hanson, "Spectral Characteristics of Medium-scale Equatorial F Region Irregularities," J. Geophys. Res., **86**, 2421, April 1981 (copy in the SSIES Project Workbook).

t. G. Knuge, Calculation of Field Lines and the Shell Parameter L from Multipole Expansions of the Geomagnetic Field, ESOC International Note No. 66, 1970.

u. VanZandt, T.E., W.L. Clark, and J.M. Warwick, "Magnetic Apex Coordinates: A Magnetic Coordinate System for the Ionospheric F2 Layer," J. Geophys. Res., **77**, 2406, May 1972.

v. Rino, C.L., "A Power Law Phase Screen Model for Ionospheric Scintillation, 1. Weak Scatter," Radio Sci., **6**, 1135, November 1979.

w. Bergland, G.D. and M.T. Dolan, "Fast Fourier Transform Algorithm," in Programs for Digital Signal Processing, IEEE Press, IEEE ISBN 0-87942-128-2, 1979.

1.3 Terms and Abbreviations.

AFGL	-	Air Force Geophysics Laboratory (former name of PL/GP)
PHG	-	Space Physics Division (former name of PL/GPSG)
AFGWC	-	Air Force Global Weather Central
DO	-	Operations Division
DOX	-	Plans Branch
SD	-	Software Development Division
SDDE	-	Space and Electromagnetic Systems Section
SDMD	-	Database Section
SDMS	-	Satellite Processing Section
WS	-	Special Support Division
WSE	-	Space Environment Support Branch
AFSFC	-	Air Force Space Forecast Center
AGDB	-	Astrogeophysical Data Base
Apex Coordinates	-	(See modified Apex coordinates)
AWS	-	Air Weather Service
C_k	-	Irregularity strength parameter
C_kL	-	Height-integrated irregularity strength parameter
Data Frame	-	A 1 second telemetry data record from the DMSP data stream
DM	-	SSIES ion Drift Meter sensor
DMSP	-	Defense Meteorological Satellite Program
EDR	-	Environmental Data Record
EDR File	-	File IESEDRFILE
EP	-	SSIES Electron Probe
FDP	-	Flight Data Processing
Fortran 77	-	ANSI Standard X3.9-1978 Fortran
F8	-	DMSP Flight 8, Block 5D/2
HSP	-	Hours Summary Print output
Modified Apex Coordinates	-	Geomagnetic latitude, longitude, and local time coordinate system (see reference 1.2.3o)
MP	-	SSIES on-board microprocessor
N_e	-	Electron density (el/cm^3)
N_i	-	Total ion density (ion/cm^3)
N[H+]	-	Density of H+ ions (ion/cm^3)
N[He+]	-	Density of He+ ions (ion/cm^3)
N[O+]	-	Density of O+ ions (ion/cm^3)
NWRA	-	Northwest Research Associates, Inc.
P_1	-	Slope of the irregularity power spectrum (one dimensional)
PSD	-	Power spectral density ((el/cm^3)^2/Hz)
PREP File	-	Mission sensor preprocessed data file containing data frames and satellite ephemeris data (IESPREPFILE)
REV	-	One readout of DMSP flight data (nominally 101 minutes)
RMS_j	-	Output from jth filter (1-9) of the SM sensor
RPA	-	SSIES ion Retarding Potential Analyzer
SDF	-	Spectral Density Function

SENPOT	-	SSIES Potential Sensor
SM	-	SSIES ion Scintillation Meter
SMC	-	Air Force Space and Missile Command (USAF Materiel Command)
CIE	-	DMSP Program Office, Engineering Division
SSIE	-	Mission Sensor for Ions and Electrons (DMSP F2 - F7)
SSIES	-	DMSP Mission Sensor for Ions, Electrons, and scintillation (DMSP spacecraft F8 - F10)
SSIES2	-	DMSP Mission Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F11 - F15)
SSIES2A	-	SSIES2 with uplinked program to output 80 telemetry data words instead of 84
SSIES3	-	DMSP Mission Sensor for Ions, Electrons, and Scintillation (DMSP spacecraft F16 - F20)
T_e	-	Electron temperature (°K)
T_i	-	Ion temperature (°K)
T_1	-	Irregularity power spectral density at a fluctuating frequency of 1 Hz
TEC	-	Total Electron Content
Transfer File	-	One of two files used to transfer processed SSIES data for storage into the AGDB (files IESAGDBXFR1 and IESAGDBXFR2)
u_h	-	Ion drift velocity, horizontal cross-track component (m/s)
u_r	-	Ion drift velocity, horizontal along-track (ram) component (m/s)
u_v	-	Ion drift velocity, vertical cross-track component (m/s)
V_{sc}	-	Spacecraft potential (volts)
$\Phi_{\Delta N}(f)$	-	Plasma irregularity spectral density function

Note: The tacit assumption that the ionospheric plasma is neutral, i.e. $N_e=N_i$, is made throughout this document.

SECTION 2. SYSTEM DESCRIPTION

2.1 System Application.

The primary objective of the SSIES data frame processor, program APGA, is to convert the engineering parameters such as currents and voltages contained in the raw data into geophysical parameters which specify the state of the earth's ionosphere as observed at the location of the DMSP spacecraft. Specifically, the following *in situ* parameters will be calculated:

a. Electron density, N_e.

b. Total ion density, N_i.

c. Heavy ion density, N[O+].

d. Light ion density, N[H+] or N[He+].

e. Electron temperature, T_e.

f. Ion temperature, T_i.

g. Ion drift velocity, u_r, u_v, and u_h.

h. Irregularity spectrum parameters $\phi_{\Delta N}(f)$, T_1, p_1, and C_kL.

The APGA, BNBA, and LDCON02 programs are written in ANSI Standard X3.9-1978 Fortran, known as Fortran 77. All of the programs have been written specifically for, or adapted to, operation on a Digital Equipment Corporation VAX computer system. The LDCON02 program uses no VAX-specific logic or routines (although filenames are specific to the VAX implementation), and could be migrated to another system with a minimum of modifications. The APGA program utilizes a small number of system routines for date and time operations. The BNBA program, which performs unpacking and bit manipulation operations specific to the VAX internal data storage configuration, **will not** migrate to another system without extensive modifications.

2.2 Security.

The APGA, BNBA, and LDCON02 programs are unclassified.

2.3 General Description.

The APGA program is executed as part of the system which processes data from the DMSP mission sensors. Figure 1 illustrates the data flow through the DMSP SSIES processing system. Figure 2 shows the files used by the APGA program.

The APGA program will process either all data available from the SSIES sensors or user selected subsets. When the program runs, it will process reformatted DMSP SSIES flight data stored in file IESPREPFILE. A subset of the data in the EDRs will be stored in one of two files (IESAGDBXFR1 or IESAGDBXFR2). This was formerly for transfer to the AGDB at AFGWC, but this operation is not implemented at SFC.

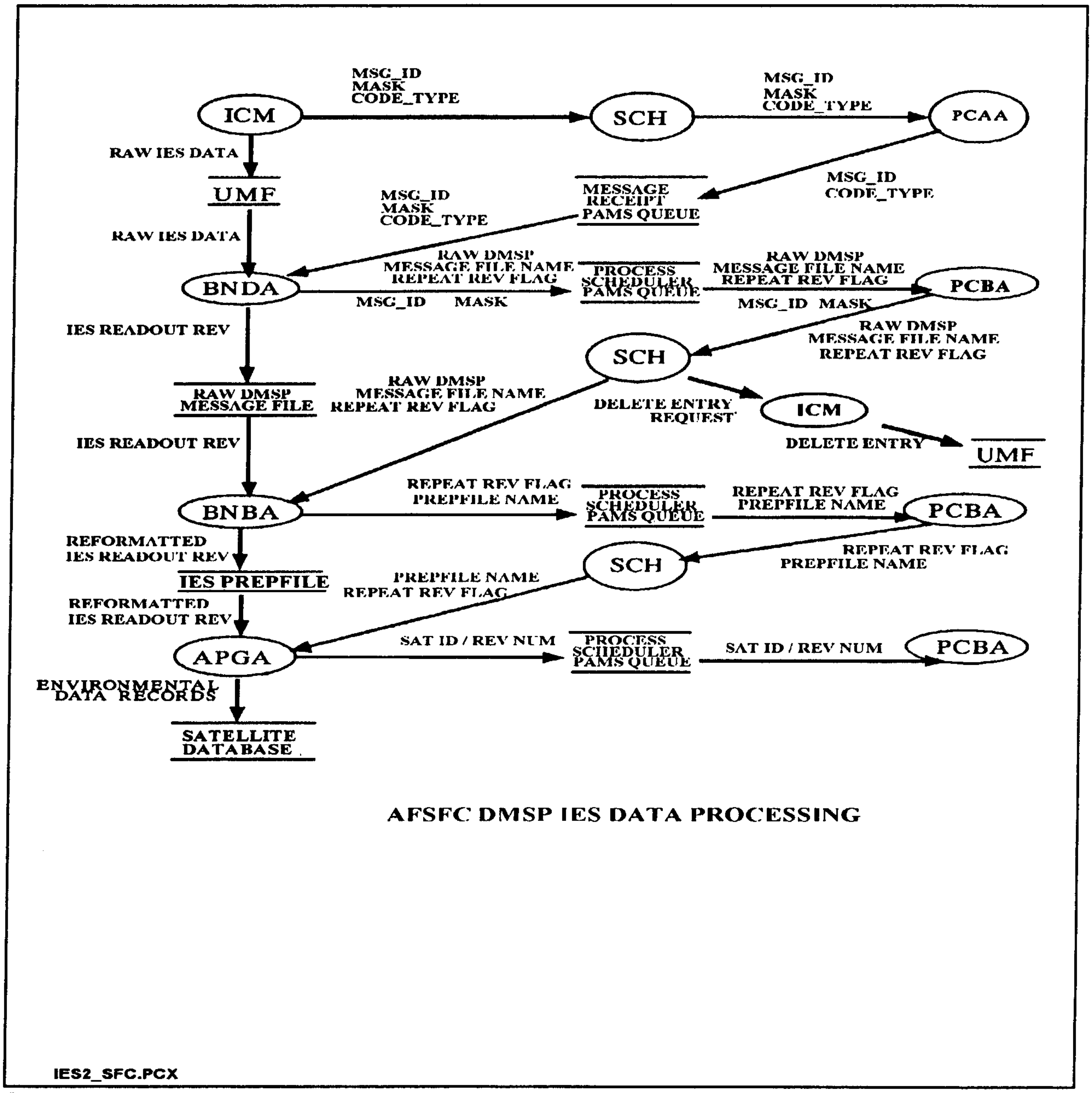


Figure 1 DMSP and APGA data flow

2.4 Program APGA Description.

Figure 3 shows the overall layout of the APGA program in hierarchy form, and Figure 4 illustrates the flow of control and data through the program. Appendix A contains structure charts for the APGA program showing the subroutine interfaces in more detail. Figure 5 is a COMMON BLOCK/SUBROUTINE cross-reference chart for the APGA program. Appendix B contains descriptions of the processing algorithms for the various sensors and data types which are implemented in the APGA program.

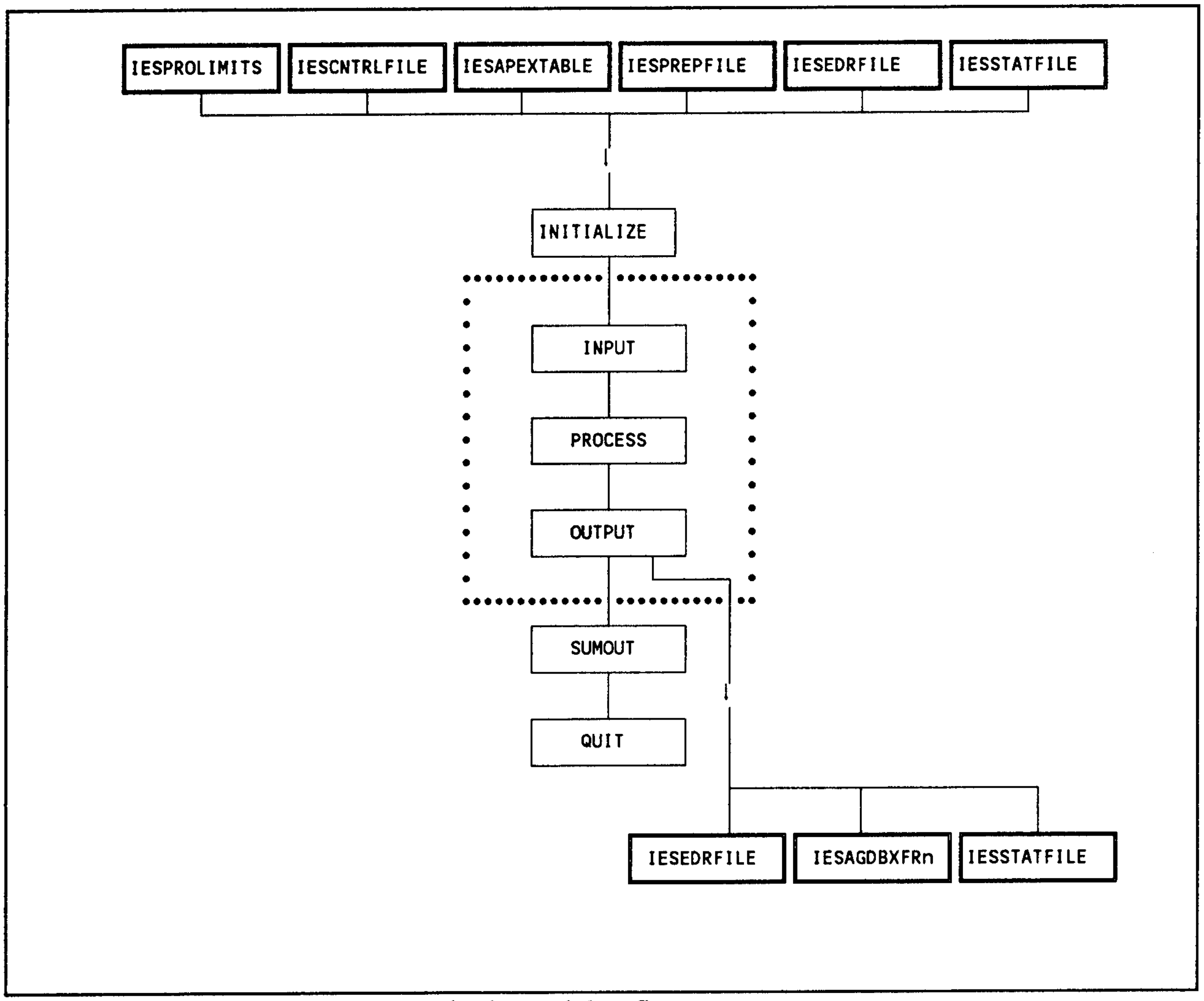


Figure 2 APGA program internal organization and data flow

APGA:	Top-level program, calling other main modules
. **INIT:**	initialize reference variables and data files
. **INPUT:**	acquire and distribute SSIES-2 data
. **PROCES:**	control processing of one data frame
. . **SMPRC:**	process scintillation meter (SM) data
. . **EPPRC:**	process electron probe (EP) data
. . **RPAPRC:**	process retarding potential analyzer (RPA) data
. . **DMPRC:**	process driftmeter (DM) data
. . **MPPRC:**	process microprocessor (MP) data
. . **CKLPRC:**	perform calculations for CKL analysis
. . **QCPRC:**	perform and report comparisons of processor results
. **OUTPUT:**	generate IESEDRFILE and IESAGDBXFR records
. **SUMOUT:**	generate summaries at end of processing
. **QUIT:**	prepare printing status and data file transfer

Figure 3 APGA program hierarchy chart

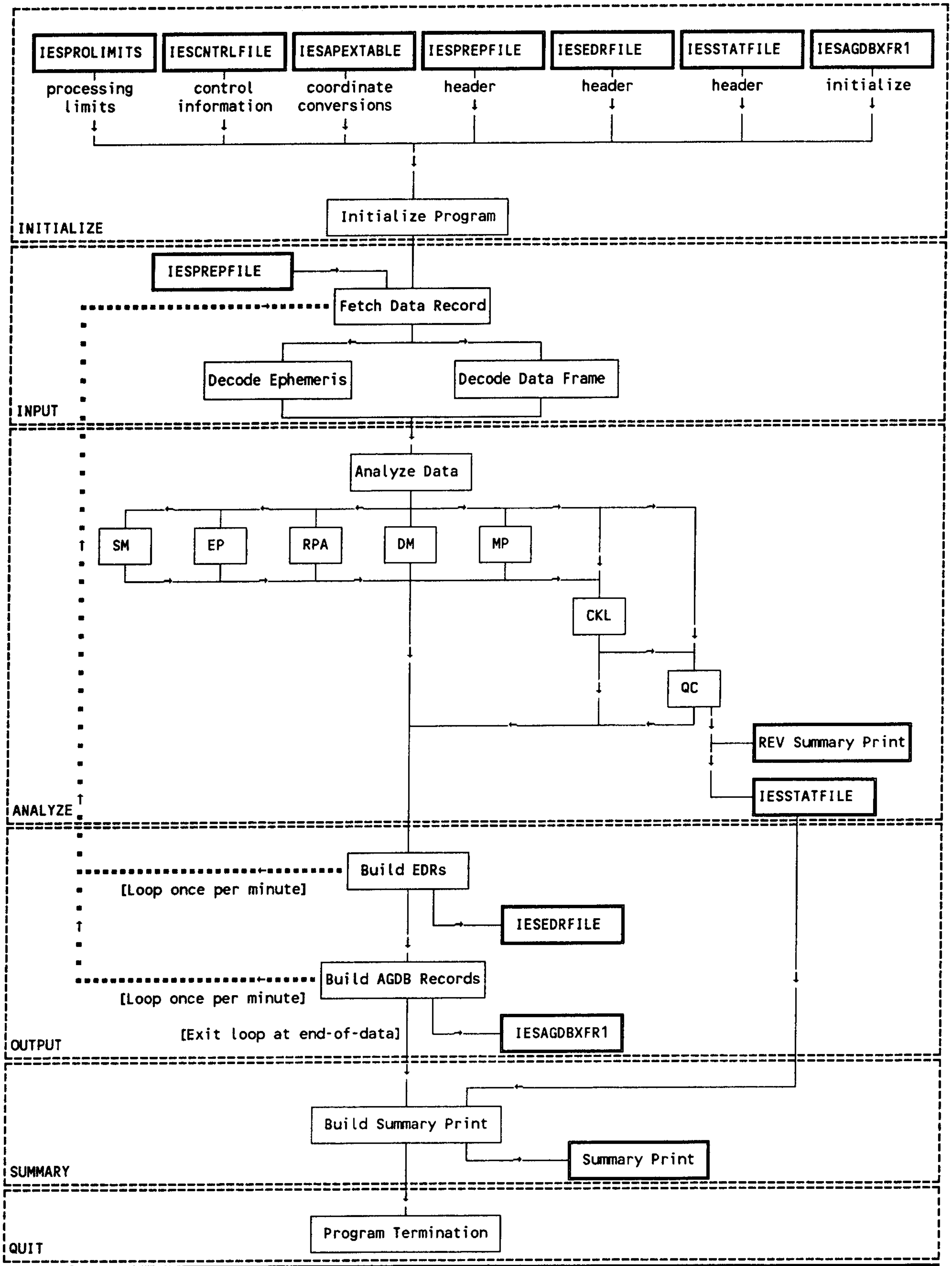


Figure 4 APGA program control and data flow

ROUTINE NAME → / COMMON BLOCK ↓↓	ADATE	ANLSAV	APXTAB	AVRAGE	BITON	BLDPDS	BSMOO	CDATE	CHEKIT	CHKSWP	CK	CKDIAG	CKLPRC	CKLPRP	CKLSAV	CMAT	COMPAR	COPY	CTRANS	CVEFF	DECDMP	DECODE	DENFIX	DENPDS	DETRND	DMCMD	DMDEN	DMDIAG	DMFIBA	DMINFO
APXLUT			X																											
CFINFO																														
CKLANL												X	X																	
CKLCON												X	X																	
CONSTS																														
DFCON																						X								
DFRAME													X								X	X								
DMANL																												X	X	
DMCON																												X	X	
EDRREC													X																	
EDRNGS																														
EDRWRK																														
EPANL													X																	
EPCON																														
IONUM																														
LUDIAG												X																X	X	
MPCON																														
MPEANL																														
MPIANL																														
PIDEG											X		X						X	X										
PREPFL																														
PROCON												X																X		
PTIMES																														
RGSTAT																										X				
RPAANL																														
RPACON										X																				
SATEPH													X																	
SATPAR																														
SCNMOD																														
SCSOH																												X		
SMANL													X																	
SMCON													X																	
STATF																														
SWPVLT																														

Figure 5 APGA program COMMON block/Subroutine cross reference

ROUTINE NAME → COMMON BLOCK ↓↓	DMPRC	DMSWCH	DMVEL	DSMSTT	DSSIES	DVAVE	EDRPRT	ELAMP	ENVMOD	EPDC	EPDIAG	EPHEM	EPPRC	EPSWP	ERF	FFT	FILERR	FILPDS	FILTER	FITLIN	FND511	FORD1	FORD2	FR2TR	FR4TR	HPMODE	HSKPNG	IAVRGE	IESPRT	IGRF
APXLUT																														
CFINFO																														
CKLANL							X																							
CKLCON																														
CONSTS	X				X								X	X																
DFCON																														
DFRAME	X										X		X														X			
DMANL	X																													
DMCON	X																										X			
EDRREC																														
EDRNGS					X																									
EDRWRK																														
EPANL										X	X		X	X																
EPCON											X		X	X																
IONUM					X																									
LUDIAG					X		X				X															X				
MPCON																														
MPEANL																														
MPIANL																														
PIDEG	X								X			X	X	X											X					
PREPFL																														
PROCON											X						X													
PTIMES																														
RGSTAT				X	X																									
RPAANL	X																													
RPACON																														
SATEPH	X											X																		
SATPAR																														
SCNMOD									X																					
SCSOH	X																										X			
SMANL																														
SMCON																														
STATF																														
SWPVLT																														

Figure 5 APGA program COMMON block/Subroutine cross reference (continued)

ROUTINE NAME → / COMMON BLOCK ↓↓	INIT	INPUT	INTERP	LATLON	LDEDR	LDSTF	LDSWPS	LDXFR	LOGLIN	LSF	MAGFLD	MEANSD	MPDIAG	MPEP	MPPRC	MPRPA	NTRP	OFFSET	OPNAPX	OPNDOF	OPNEDR	OPNPRP	OPNSTF	OPNXFR	OUTPUT	PLASMA	PRMRNG	PRNTON	PROCES	QCEP
APXLUT																		X	X											
CFINFO																					X									
CKLANL					X																									
CKLCON																														
CONSTS																										X				
DFCON																														
DFRAME					X									X	X	X									X				X	
DMANL					X																									
DMCON																														
EDRREC					X		X																		X					
EDRNGS																											X			
EDRWRK																					X									
EPANL					X								X																	
EPCON																														
IONUM	X					X													X		X		X	X						
LUDIAG													X																	
MPCON																														
MPEANL					X								X	X																
MPIANL					X								X			X														
PIDEG	X			X							X							X								X				
PREPFL		X																				X								
PROCON	X	X			X								X											X	X			X	X	
PTIMES	X																													
RGSTAT																														
RPAANL					X								X																	
RPACON																										X				
SATEPH				X	X								X												X					
SATPAR	X																													
SCNMOD	X																													
SCSOH					X																									
SMANL					X																									
SMCON																														
STATF						X																								
SWPVLT																														

Figure 5 APGA program COMMON block/Subroutine cross reference (continued)

ROUTINE NAME → / COMMON BLOCK ↓↓	QCPRC	QCRPA	QUIT	RANGE	RDPREP	RPAALG	RPAPRC	RPASWP	RPDIAG	SCDIAG	SELNRM	SETEDR	SFSUM	SMCMD	SMDIAG	SMPRC	SSLOPE	SUMOUT	SWPCOL	TANDP	TIMCON	TIMES	UPBITS	VEHPAR	VSWEEP	WINDOW	WRTEDR	WRTXFR	XFRPRT	XPOS
APXLUT																														
CFINFO												X															X			
CKLANL	X																													
CKLCON																								X						
CONSTS						X		X								X														
DFCON																								X						
DFRAME	X						X		X	X					X	X														
DMANL	X										X																			
DMCON																								X						
EDRREC												X																		
EDRNGS																														
EDRWRK												X															X			
EPANL	X						X				X																			
EPCON											X													X						
IONUM			X									X						X									X	X		
LUDIAG			X						X						X			X										X	X	
MPCON																								X						
MPEANL	X																													
MPIANL	X																													
PIDEG								X																		X				
PREPFL					X																									
PROCON	X		X						X	X					X			X						X				X		
PTIMES					X																									
RGSTAT														X																
RPAANL	X					X	X		X		X					X														
RPACON						X	X	X	X															X						
SATEPH	X						X			X						X														
SATPAR																								X						
SCNMOD																														
SCSOH	X																													
SMANL	X										X				X	X														
SMCON															X	X								X						
STATF	X																	X												
SWPVLT																			X						X					

Figure 5 APGA program COMMON block/Subroutine cross reference (continued)

2.4.1 APGA Main Routine.

a. Function.

The main routine controls processing of the DMSP SSIES data.

b. Inputs.

There are no inputs to the main routine. Other routines perform all initialization and input/output. A logical end of data status flag (variable THEEND) indicating that all data have been processed is returned to this routine by Subroutine INPUT. See the associated subroutines for further details.

c. Processing.

1. Obtain the system wall and CPU times and print them out (Subroutine TIMES).

2. Initialize the program (Subroutine INIT).

3. Fetch and process one second of raw SSIES data until all data in the PREP file have been processed as follows:

   (a) Fetch a one second data frame (Subroutine INPUT). If the end of data status flag (THEEND) is .TRUE. and no data have been previously processed, jump to step 4.

   (b) Process the data frame, converting the raw data into geophysical parameters (Subroutine PROCESS).

   (c) Build and output Environmental Data Records (EDRs) to the EDR file and summary records into a temporary file for transfer the AGDB (Subroutine OUTPUT).

   (d) Loop back to step 3(a).

4. Create any Hours Summary Print (HSP) output required (Subroutine SUMOUT).

5. Conduct program termination activities, which include closing all files used by the program, alerting the system operator of any abnormal conditions encountered in the processing which require correction, and setting the appropriate status values to control execution of the PRINTAWAY processor (not implemented at SFC) and to designate which of the two AGDB transfer files needs to be transferred to the AGDB (not implemented at SFC).

6. Obtain the system wall and CPU times and print them out (Subroutine TIMES).

7. STOP.

d. Outputs.

The main routine directly produces only one output, the system wall and CPU times at the start and end of the program run and the total number of frames of data processed which are entered into the HSP. See the associated subroutines for further details.

e. Associated subroutines.

(See Figure 3 for a hierarchy chart.)

1. Subroutine TIMES. Fetch the system wall and CPU time and write them to the HSP (see 2.4.1.1)

2. Subroutine INIT. Perform all program initialization procedures (see 2.4.2).

3. Subroutine INPUT. Interface with the SFC IESPREPFILE to fetch and decode reformatted SSIES data frames and satellite ephemeris records (see 2.4.3).

4. Subroutine PROCESS. Process the data frames to convert the raw sensor data into geophysical parameters (see 2.4.4).

5. Subroutine OUTPUT. Build and store EDRs and AGDB summary records (see 2.4.5).

6. Subroutine SUMOUT. Create all required summary HSP output (see 2.4.6).

7. Subroutine QUIT. Perform all end of processing clean up procedures (see 2.4.7).

f. Interfaces.

1. Uses no COMMON blocks.

2.4.1.1 Subroutine TIMES.

a. Function.

Fetch the current system wall and CPU times and print them out.

b. Inputs.

1. Argument list.

LABEL	-	CHARACTER message to print out with times
LUALT	-	Logical unit of alternate print file

c. Processing.

1. Fetch the current system wall time (Subroutine ADATE).

2. Fetch the current system CPU time (VAX library routine LIB$STAT_TIMER).

3. Print out the current system wall and CPU time with the message input via the argument list.

4. If an alternate print file has been requested, i.e. LUALT is non-zero, print the same message into the file designated by LUALT.

5. Return to the main routine.

d. Outputs.

The current system wall and CPU time and input message are output to HSP in the following format:

**WALL: HH:MM:SS CPU: xxxx.xxx Message in argument list

e. Associated subroutines.

None.

f. Interfaces.

1. Called from the main routine.

2. Uses no COMMON blocks.

2.4.1.2 BLOCK DATA Routine DSSIES.

a. Function.

Initialize COMMON blocks CONSTS, EDRNGS, IONUM, LUDIAG, and RGSTAT.

b. Inputs.

None.

c. Processing.

None.

d. Outputs.

1. COMMON blocks.

/CONSTS/

E	-	Charge on an electron (coulombs)
K	-	Boltzmann's constant
MP	-	Proton rest mass (kg)
ME	-	Electron rest mass (kg)

/EDRNGS/

DENMIN	-	Minimum allowed value, density (cm^{-3})
DENMAX	-	Maximum allowed value, density (cm^{-3})
IDEN	-	Out-of-range replacement flag (0 or 1)
TEMIN	-	Minimum allowed value, T_e (°K)
TEMAX	-	Maximum allowed value, T_e (°K)
ITE	-	Out-of-range replacement flag (0 or 1)
TIMIN	-	Minimum allowed value, T_i (°K)

TIMAX	-	Maximum allowed value, T_i (°K)
ITI	-	Out-of-range replacement flag (0 or l)
UHMIN	-	Minimum allowed absolute value, u_h or u_r (m/s)
UHMAX	-	Maximum allowed absolute value, u_h or u_r (m/s)
IUH	-	Out-of-range replacement flag (0 or 1)
UVMIN	-	Minimum allowed absolute value, u_v (m/s)
UVMAX	-	Maximum allowed absolute value, u_v (m/s)
IUV	-	Out-of-range replacement flag (0 or 1)
P1MIN	-	Minimum allowed value, p_1
P1MAX	-	Maximum allowed value, p_1
IP1	-	Out-of-range replacement flag (0 or 1)
T1MIN	-	Minimum allowed value, T_1
T1MAX	-	Maximum allowed value, T_1
IT1	-	Out-of-range replacement flag (0 or 1)
CKLMIN	-	Minimum allowed value, C_kL
CKLMAX	-	Maximum allowed value, C_kL
ICKL	-	Out-of-range replacement flag (0 or 1)
VLTMIN	-	Minimum allowed value, V_{sc} (volts)
VLTMAX	-	Maximum allowed value, V_{sc} (volts)
IVLT	-	Out-of-range replacement flag

/IONUM/

LUOUT	-	Logical unit for standard output print file
LUIN	-	Logical unit for standard input file
LUCF	-	Logical unit for file IESCNTRLFILE
LUAPEX	-	Logical unit for file IESAPEXTABLE
LUEDR	-	Logical unit for file IESEDRFILE
LUXFR	-	Logical unit for file IESAGDBXFRn
LUSTAT	-	Logical unit for file IESSTATFILE

/LUDIAG/

LUSMA	-	Logical unit for SMPRC ASCII diagnostic output
LUSMB	-	Logical unit for SMPRC binary diagnostic output
LUMPEP	-	Logical unit for file IESMPEPPRT
LUMPRP	-	Logical unit for file IESMPRPAPRT
LUEPA	-	Logical unit for EPPRC ASCII diagnostic output
LUEPB	-	Logical unit for EPPRC binary diagnostic output
LURPAA	-	Logical unit for RPAPRC ASCII diagnostic output
LURPAB	-	Logical unit for RPAPRC binary diagnostic output
LUDMA	-	Logical unit for DMPRC ASCII diagnostic output
LUDMB	-	Logical unit for DMPRC binary diagnostic output
LUDMHP	-	Logical unit for DM H+ mode diagnostic output
LUCKLA	-	Logical unit for CKLPRC ASCII diagnostic output
LUCKLB	-	Logical unit for CKLPRC binary diagnostic output
LUQCA	-	Logical unit for QCPRC ASCII diagnostic output
LUEDRA	-	Logical unit for WRTEDR ASCII diagnostic output
LUXFRA	-	Logical unit for WRTXFR ASCII diagnostic output

/RGSTAT/

IDMSTA	-	Driftmeter mode status
IWGSTA	-	Wiggle level status
IRPSTA	-	Repeller level status
IVBSTA	-	VBias/SENPOT status
IW1STA	-	WIBAN1 Range status
IW2STA	-	WIBAN2 Range status
IRFSTA	-	SENPOT relay flag status
ISTSTA	-	Subcom processing status

e. Associated subroutines.

None.

f. Interfaces.

1. Uses COMMON blocks CONSTS, EDRNGS, IONUM, LUDIAG, and RGSTAT.

2.4.2 Subroutine INIT.

a. Function.

Open and, when necessary, initialize all non-diagnostic files used by the program, and initialize variables in COMMON blocks PIDEG, PROCON, PTIMES, SATPAR (from file IESCNTRLFILE), and SCNMOD (also from file IESCNTRLFILE). Figure 6 is a hierarchy chart for this module.

b. Inputs.

1. Files.

(a) IESCNTRLFILE. All parameters in COMMON blocks SATPAR and SCNMOD.

(b) IESPROLIMITS. This test file is used to input constraints on the amount of raw data to be processed. If this file is not found, or is empty, there will be no limits placed on the processing. The inputs from this file are the day of year (JDAY1) and time of day (IPTIM1) (seconds since midnight, GMT) to start processing data and the number of seconds to process (NSECSP). These parameters are passed to the input routines via COMMON PTIMES.

NOTE: THIS OPTION IS FOR TEST PURPOSES ONLY - FILE IESPROLIMITS SHOULD NOT BE ASSIGNED AS PART OF THE PRODUCTION APGA RUNSTREAM.

c. Processing.

1. Calculate the trigonometric constants in COMMON PIDEG.
(Note: These are all based on the calculation of $\pi/2$ from the arctangent of 1.)

2. Initialize the processor error flag (ICNERR), console message disable flag (ICNWRT), transfer file flag (NFXFR), and current system wall time variables (IRDATE, IRTIME, and IRMIN) in

INIT:	initialize reference variables and data files
. CDATE:	reformat date and time for program use
. . ADATE:	date and time report from system
. . TIMCON:	convert between UT and IES reference minutes
. VSWEEP:	initialize reference EP and RPA sweep voltages
. OPNPRP:	open IESPREPFILE and read RIR record
. . FILERR:	determine and report file error status
. VEHPAR:	acquire processing parameter values for current satellite
. . COPY:	initialize array or transfer values between arrays
. OPNEDR:	initialize IESEDRFILE
. . PRNTON:	set PRINTAWAY status for conclusion of processing
. . FILERR:	determine and report file error status
. OPNAPX:	acquire APEX conversion table from IESAPEXTABLE
. . FILERR:	determine and report file error status
. OPNXFR:	initialize IESAGDBXFR1 or IESAGDBXFR2
. . PRNTON:	set PRINTAWAY status for conclusion of processing
. . FILERR:	determine and report file error status
. OPNSTF:	initialize IESSTATFILE
. . PRNTON:	set PRINTAWAY status for conclusion of processing
. . FILERR:	determine and report file error status
. FILERR:	determine and report file error status

Figure 6 INIT module hierarchy chart

COMMON PROCON.

Note: The flag ICNWRT is also input from file IESCNTRLFILE. It is set to zero, which enables the console message option, in case an error is encountered in reading the control parameters from IESCNTRLFILE.

3. Input processing control parameters from test file IESPROLIMITS. If no file or inputs are found, JDAY1 is set to -1 to disable this option.

   Note: In normal production mode, JDAY1 should always be -1

4. Read in the program control parameters from file IESCNTRLFILE and load COMMON blocks SATPAR and SCNMOD.

5. Open/initialize all system files as follows:

   (a) File IESPREPFILE which contains the raw SSIES data frames (Subroutine OPNPRP).

   (b) File IESEDRFILE to which EDRs are written (Subroutine OPNEDR).

   (c) File IESAPEXTABLE which contains the information for calculating geomagnetic Apex coordinates (Subroutine OPNAPX).

   (d) File IESAGDBXFR1 or IESAGDBXFR2 to which EDR summary records will be written for later transfer to AGDB (not implemented at SFC) (Subroutine OPNXFR).

   (e) File IESSTATFILE to which REV-averaged processing statistics are written (Subroutine

OPNSTF).

6. If any errors are encountered, an error message is added to the HSP, the PRINTAWAY disable flag is set, the console error message flag is set (all in Subroutine FILERR), and the return status flag is set to -1, indicating an error condition was encountered. If no errors were encountered, this flag is set to 0.

7. Return to the main routine.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. COMMON blocks.

/PIDEG/

PI	-	Value of π
TWOPI	-	Value of 2π
PIBY2	-	Value of $\pi/2$
RAD	-	Conversion factor, degrees to radians
DEG	-	Conversion factor, radians to degrees
RADHR	-	Conversion factor, hours to radians
HOUR	-	Conversion factor, radians to hours

/PROCON/

IPAWAY	-	PRINTAWAY disable flag
INTSUM	-	IESSTATFILE summary print interval (hours)
ISSDGP	-	Processing control diagnostic print flag
ICNWRT	-	Console message disable flag
ICNERR	-	Console message error flag
NFXFR	-	Transfer file number (always 1)

IRDATE	-	Date of current run (YYMMDD, GMT)
IRTIME	-	Time of current run (HHMMSS, GMT)
IRMIN	-	Time of current run (IES minutes)

/PTIMES/

JDAY1	-	Julian day to start processing (test only)
IPTIM1	-	Time to start processing (test only)
NSECP	-	Number of seconds of data to process (test only)

/SATPAR/

NUMSAT	-	Number of active satellites in CNTRLFILE
MISSN	-	Active DMSP satellite Mission IDs
IDFLT	-	Active DMSP satellite Flight IDs

INFVEH - Processing control parameters from CNTRLFILE

/SCNMOD/

AE	-	Equatorial model value for a
AA	-	Auroral model value for a
BH	-	Auroral model value for b
GMLA	-	Equatorial region boundary (deg)
GMLAW	-	Equatorial region transition width (deg)
GML1	-	Auroral boundary model lead term (deg)
CK	-	Variation of auroral boundary with K_p
CBT	-	Variation of auroral boundary with time
DBT	-	Phase of auroral boundary time variation (hours)
CHB	-	Auroral region transition width parameter
DELTA0	-	Model value for angle between magnetic L-shell and irregularity sheets
LE	-	Equatorial model value for L
LM	-	Mid-latitude model value for L
LA	-	Auroral model value for L
Q	-	Model value for q

e. Associated subroutines.

(See Figure 6 for a hierarchy chart).

1. Subroutine CDATE. Obtain the current system wall time (see 2.4.2.1).

2. Subroutine VSWEEP. Calculate and store sweep voltages for all EP and RPA sweep types (see 2.4.2.2).

3. Subroutine OPNPRP. Prepare file IESPREPFILE for processing (see 2.4.2.3).

4. Subroutine VEHPAR. Load proper processor control information for the proper DMSP vehicle (see 2.4.2.4).

5. Subroutine OPNEDR. Prepare file IESEDRFILE for processing (see 2.4.2.5).

6. Subroutine OPNAPX. Input the parameters and tables used to calculate geomagnetic Apex coordinates from file IESAPEXTABLE (see 2.4.2.6).

7. Subroutine OPNXFR. Determine which transfer file should be used (IESAGDBXFR1 or IESAGDBXFR2) and prepare it for processing (see 2.4.2.7).

8. Subroutine OPNSTF. Prepare file IESSTATFILE for processing (see 2.4.2.8).

9. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

f. Interfaces.

1. Called from the main routine.

2. Uses COMMON blocks IONUM, PIDEG, PROCON, PTIMES, SATPAR, and SCNMOD.

2.4.2.1 Subroutine CDATE.

a. Function.

Obtain the current system wall time, reformat it, and calculate the IES minute corresponding to that time.

b. Inputs.

None.

c. Processing.

1. Obtain the system wall time (Subroutine ADATE).

2. Reformat the date from MMDDYY in CHARACTER*8 format to YYMMDD in INTEGER, and the time from HHMMSS in CHARACTER*8 to HHMMSS in INTEGER.

3. Calculate the IES minute corresponding to the current wall time (Subroutine TIMCON). (Note: The IES minute is the primary database time used by this program. It is (somewhat arbitrarily) defined as the number of minutes elapsed since 0000 GMT on 1 January 1985. See section 2.4.8.12 on Subroutine TIMCON for more details.)

4. Return to Subroutine INIT.

d. Outputs.

1. Argument list.

ICDATE	-	Current date (YYMMDD, integer)
ICTIME	-	Current time (HHMMSS, integer)
ICMIN	-	Current IES minute (integer)

e. Associated subroutines.

1. Subroutine TIMCON. Calculate the IES minute for the current date/time (see 2.4.8.12).

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses no COMMON blocks.

2.4.2.2 Subroutine VSWEEP.

a. Function.

Calculate the 'ideal' sweep voltages for all types of EP and RPA sweeps. This is necessary because, beginning with the SSIES-2 configuration (satellite F-11), the EP and RPA sweep voltages are not included in the downlinked telemetry data.

b. Inputs.

1. Argument list.

ISSDGP	-	Diagnostic print control switch

c. Processing.

1. Establish the seconds counter loop (K = 1 to 4).

2. Calculate the seconds-dependent (K) constants.

3. Establish the value pointer within a second loop (J = 1 to 25).

4. Calculate the J-dependent constants.

5. Calculate the up and down sweep voltage values for RPA and the normal (4 second) EP sweeps.

6. During the first 3 iterations of the seconds index, calculate up and down sweep voltage values for the EP bias or calibration (3 second) sweeps.

7. During the first 2 iterations of the seconds index, calculate the up and down sweep voltage values for the EP Mode E short (2 second) sweeps.

8. If the diagnostic print switch is set (> 0) print the calculated voltage values to the HSP.

9. Return to Subroutine INIT.

d. Outputs.

1. COMMON blocks.

/SWPVLT/

SVOLTS	-	Array of calculated sweep voltages

NOTE: SVOLTS(I, J, K) is arranged such that:

I	= (1,2,....25)	points to the sweep voltages within a given second.
J	= (1)	points to down sweeps.
	= (2)	points to up sweeps.
K	= (1,..,4)	points to the K^{th} second of the normal EP sweep.
K	= (5, 6, 7)	points to the $(K-4)^{th}$ second of the EP bias sweep
K	= (8, 9)	points to the $(K-7)^{th}$ second of the EP short sweep
K	= (10,..,13)	points to the K^{th} second of the RPA sweep.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON block SWPVLT.

2.4.2.3 Subroutine OPNPRP.

a. Function.

Open the designated IESPREPFILE and read the Readout Information Record (RIR).

b. Inputs.

1. Argument list.

ISSDGP	-	Diagnostic print switch

2. Files.

The first 10 words of the RIR record are read in from the IESPREPFILE.

c. Processing.

1. Open the designated IESPREPFILE.

2. If the OPEN was successful, read the first record of the IESPREPFILE and store the first 10 words of the RIR into COMMON block /PREPFL/.

3. Set the NFRAME pointer to 60 to signal Subroutine RDPREP that a 1 minute data record needs to be read from the IESPREPFILE.

4. If an OPEN error was encountered, invoke Subroutine FILERR to set the necessary error statuses and set the return status flag to -1. If no error is encountered, this status is set to 0.

4. Return to Subroutine INIT.

d. Outputs.

1. Argument list.

IDMISS	-	Mission ID from RIR data
ISTAT	-	Return status

2. COMMON blocks.

/PREPFL/

RIR	-	10 words of RIR data
NFRAME	-	Telemetry data frame pointer

e. Associated subroutines.

1. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON block PREPFL.

2.4.2.4 Subroutine VEHPAR.

a. Function.

Load the proper processor control information and instrument constants for the DMSP spacecraft whose data are being processed.

b. Inputs.

1. Argument list.

IDMISS	-	Satellite mission ID
ISSDGP	-	Diagnostic print switch

2. COMMON blocks.

/SATPAR/

IDFLT	-	Satellite flight ID
INFVEH	-	Vehicle processor control and instrument constant information block
MISSN	-	Valid mission IDs
NUMSAT	-	Number of satellites for which control information is available

3. Files.

None.

c. Processing.

1. Compare the current satellite ID to the list of satellite IDs contained in file IESCNTRLFILE. This information was loaded into COMMON block SATPAR during program initialization.

2. If the satellite information is available for the requested satellite, load the information into the processor control COMMON blocks.

3. Print a diagnostic message if the satellite ID is not contained in IESCNTRLFILE.

4. Return a status flag indicating an error if the satellite was not contained in the file.

d. Outputs.

1. Argument list.

ISTVP	-	Subroutine action status flag

2. COMMON blocks.

/PROCON/

IFLT	-	Flight ID
INFODG	-	Diagnostic print flags
INFOPR	-	Processor action control flags
MISSID	-	Satellite mission ID

/SMCON/

INFOSM	-	SM processor constants

/EPCON/

INFOEP	-	EP processor constants

/RPACON/

INFORP	-	RPA processor constants

/DMCON/

INFODM	-	DM processor constants

/CKLCON/

INFOCK	-	CKL processor constants

/MPCON/

INFOMP	-	MP processor constants

/DFCON/

FCONV	-	OLS data word conversion constants

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON blocks SATPAR, PROCON, SMCON, EPCON, RPACON, DMCON, CKLCON, MPCON, and DFCON.

2.4.2.5 Subroutine OPNEDR.

a. Function.

Open and initialize file IESEDRFILE.

b. Inputs.

1. Argument list.

NUMSAT	-	Number of active satellites
IDFLT	-	Flight IDs of the active satellites

2. COMMON blocks.

/IONUM/

LUEDR	-	Logical unit for IESEDRFILE

3. Files.

IESEDRFILE. The header block for the first satellite stored in the file is input. See 3.3.2.2 for a description of the contents of the header block.

c. Processing.

1. Open file IESEDRFILE.

2. If the OPEN was successful, set the return status flag to zero and reinitialize the file as follows:

(a) Write out a warning that the file is being reinitialized, disable the PRINTAWAY processor, and set the console message flag to indicate that the HSP should be examined (Subroutine PRNTON).

(b) Zero out the processing status flag array (IBPFLG).

(c) Set up an initial header array for each satellite specified by IESCNTRLFILE and write it (with the zeroed IBPFLG array) out to file IESEDRFILE.

NOTE: The necessity for the initialization procedure is a remnant of the AFGWC system, and is

done to accommodate the logic built in to the EDR output subroutines. When the EDR output subroutines are simplified, the initialization procedure can be eliminated.

4. If an error was encountered, invoke Subroutine FILERR to set the necessary error statuses and set the return status flag to -1.

5. Return to Subroutine INIT.

d. Outputs.

1. Argument list.

ISTAT	-	Return status flag

2. COMMON blocks.

/CFINFO/

IENREC	-	Number of EDR I/O blocks per satellite
IENSAT	-	Number of active satellites

3. Files.

If file IESEDRFILE was initialized, header blocks are written to the file for each active satellite specified by file IESCNTRLFILE.

e. Associated subroutines.

1. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

3. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON blocks CFINFO, EDRWRK, and IONUM.

2.4.2.6 Subroutine OPNAPX.

a. Function.

Open file IESAPEXTABLE and read the Apex coordinate conversion table information into COMMON APXLUT.

b. Inputs.

1. COMMON blocks.

/IONUM/

LUAPEX - Logical unit for file IESAPEXTABLE

2. Files.

The entire contents of file IESAPEXTABLE are read in. See 3.3.2.8 for the contents.

c. Processing.

1. Open the file.

2. Read in the contents.

3. Close the file.

4. If an error was encountered, invoke Subroutine FILERR to set the necessary error statuses and set the return status flag to -1. If no error is encountered, this status is set to 0.

5. Return to Subroutine INIT.

d. Outputs.

1. Argument list.

ISTAT - Return status

2. COMMON blocks.

/APXLUT/

ALTAPX	-	Altitude for which the information is valid
XO,YO,ZO	-	Offset components of offset dipole system
UO	-	Rotation matrix for offset dipole system
TABLE	-	Apex coordinate look-up table

e. Associated subroutines.

1. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON blocks APXLUT and IONUM.

2.4.2.7 Subroutine OPNXFR.

a. Function.

Open AGDB summary record transfer file 1.

b. Inputs.

1. COMMON blocks.

/IONUM/

LUXFR	-	Logical unit for file IESAGDBXFRn

/PROCON/

IRDATE	-	Current date (YYMMDD)
IRTIME	-	Current time (HHMMSS)

2. Files.

None

c. Processing.

1. Set the transfer file number to 1.

2. Open transfer file 1.

3. If the OPEN was successful, set the lock flag to 99 and write lock flag, current date, and current time to the first record of the file, and set the return status to zero.

4. Write a diagnostic message to HSP designating that file 1 was selected.

5. If an error was encountered set the return status flag to -1.

6. Return to Subroutine INIT.

NOTE: The transfer files are a remnant of AFGWC processing, and are probably of no use for SFC processing purposes.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. COMMON blocks.

/PROCON/

NFXFR	-	Transfer file used (always 1)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON blocks IONUM and PROCON.

2.4.2.8 Subroutine OPNSTF.

a. Function.

Open and initialize file IESSTATFILE.

b. Inputs.

1. Argument list.

NUMSAT	-	Number of active satellites
MISSN	-	Mission IDs of the active satellites
IDFLT	-	Flight IDs of the active satellites

2. COMMON blocks.

/IONUM/

LUSTF	-	Logical unit for file IESSTATFILE

3. Files.

None.

c. Processing.

1. Open file IESSTATFILE.

2. If the OPEN was successful, initialize the file as follows:

(a) Write out the header information for each of the three satellite sections of the file with the Mission ID and Flight ID for each active satellite, with the last update (LSTUPD) and last displayed (LSTDIS) times set to -1 and the number of REVs flag (NREVS) set to zero (0).

(b) Write out a warning that the file is being reinitialized, disable the PRINTAWAY processor, and set the console message flag to indicate that the HSP should be examined (Subroutine PRNTON).

3. If an error was encountered, invoke Subroutine FILERR to set the necessary error statuses and set the return status flag to -1. If no error is encountered, this status is set to 0.

4. Return to Subroutine INIT.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. Files.

If file IESSTATFILE was initialized, header blocks are written to the file for each active satellite specified by file IESCNTRLFILE.

e. Associated subroutines.

1. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

2. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine INIT.

2. Uses COMMON block IONUM.

2.4.3 Subroutine INPUT.

a. Function.

Control the flow of data to the individual instrument processors. Figure 7 is a hierarchy chart for this module.

INPUT:	acquire and distribute SSIES-2 data
. RDPREP	acquire a one minute data record from IESPREPFILE
. EPHEM:	prepare ephemeris information for processing
.. TIMCON:	convert between UT and IES reference minutes
.. LATLON:	calculate latitude and longitude from orbit parameters
.. APXTAB:	convert to APEX latitude and longitude
... OFFSET:	convert between geographic and magnetic dipole coordinates
... INTERP:	interpolate from a list of values
.. PRNTON:	set PRINTAWAY status for conclusion of processing
. CHEKIT:	assess validity of one-second data frame
. DECODE:	distribute data frame values for processing, in proper units
.. DSMSTT:	determine instrument status from DSM subcom
.. DECDMP:	print data frame information, for diagnostics
. HSKPNG:	acquire and assess sensor housekeeping data

Figure 7 INPUT module hierarchy chart

b. Inputs.

1. Argument list.

   None.

2. COMMON blocks.

   /PROCON/

ISSDGP	-	Diagnostic print switch

   /PREPFL/

IDATSEC	-	Array of raw data frame times
IPHBLK	-	Array of ephemeris data values
IRAWVAL	-	Array of raw data values
NFRAME	-	Pointer to current data frame

3. Files.

   None.

c. Processing.

One minute data records are retrieved from the IESPREPFILE. One second segments of the record are decoded, and each instrument processor is given the opportunity to apply the data. Some of the individual instrument processors must keep counters and indices updated, thus requiring that a zero data frame be passed around for missing data seconds. In addition, some processors can handle only a few missing seconds without having to reinitialize. As a result, several flags are provided indicating current input status.

The duplicate frame flag is the result of a problem identified during hardware testing. Due to timing within the OLS sequence of events, a duplicate data frame can be received in the data stream. Space Division has requested that the occurrence be identified and a count summary provided.

1. If the frame pointer NFRAME is 60, the previous 1 minute data record has been processed. Invoke Subroutine RDPREP to read the next 1 minute data record. If NFRAME is less than 60, go to step 3.

2. If the read was successful, set the frame pointer to zero and invoke Subroutine EPHEM to decode the satellite ephemeris information accompanying the one minute PREPFILE record. If the ephemeris record is bad, set the general reset flag and read to the next record. If there are successive (10 or more) bad ephemeris records, set the termination flag to end the run.

3. Increment the frame pointer and invoke Subroutine CHEKIT to determine that data frame validity.

4. If the data are not valid and:

   a. this is the beginning of the run, go back to step 3.

b. this is not the beginning of the run, pass a zero frame through the instrument processing routines.

5. If the data are valid, invoke Subroutine DECODE to convert the raw data into engineering parameters.

6. Invoke Subroutine HSKPNG to update the sensor housekeeping data.

7. If the diagnostic print switch is set, write diagnostic output to the standard output file.

8. Return to the main routine.

d. Outputs.

1. Argument list.

THEEND	-	End of all unprocessed data flag

2. COMMON blocks.

/PROCON/

GRESET	-	General reset indicator flag for the instrument processors
NEWREV	-	Data from different readout indicator flag
NEWSAT	-	Data from different satellite indicator flag
NUMREV	-	Readout REV number

3. Files.

None.

e. Associated subroutines.

1. Subroutine RDPREP. Read the next record from the IESPREPFILE (see 2.4.3.1).

2. Subroutine EPHEM. Provide vehicle ephemeris and orbital parameters (see 2.4.3.2).

4. Subroutine CHEKIT. Determine the validity of the data frame (see 2.4.3.3).

5. Subroutine DECODE. Convert raw telemetry data to engineering parameters (see 2.4.3.4).

6. Subroutine HSKPNG. Update sensor housekeeping data (see 2.4.3.5).

f. Interfaces.

1. Called from main program.

2. Uses COMMON block PROCON.

2.4.3.1 Subroutine RDPREP.

a. Function.

Read a one minute data record from the IESPREPFILE.

b. Inputs.

1. Argument list.

None

2. COMMON blocks.

/PREPFL/

NODYMD	-	Packed year, month, and day of last ascending node from RIR data

/PTIMES/

JDAY1	-	Julian day of interest for limited processing
IPTIM1	-	Start time in seconds for limited processing
NSECSP	-	Number of seconds of limited processing

3. Files.

A data record is read from the IESPREPFILE.

c. Processing.

1. Read the next sequential record of the IESPREPFILE.

2. If the read was not successful, set the termination flag (THEEND) to .TRUE., write the I/O status to the print file, and return to Subroutine INPUT.

3. If the read was successful:

   a. store the packed nodal year, month, and day from the RIR record into the 28th word of the ephemeris block.

   b. If limited processing was selected and the data record time is lower than the start time, go back to step 1.

   c. If limited processing was selected and the data record time is greater than the end time, set the termination flag to .TRUE. and return to Subroutine INPUT.

4. Return to Subroutine INPUT.

d. Outputs.

1. Argument list.

THEEND	-	Termination flag
IRDSTAT	-	VAX I/O status value from read operation

2. COMMON blocks.

/PREPFL/

IDATSEC	-	Array of one second data frame times
IPHBLK	-	Ephemeris data block
IRAWVAL	-	Array of raw data frame values
NFRAME	-	Pointer to current data frame

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INPUT.

2. Uses COMMON blocks PREPFL and PTIMES.

2.4.3.2 Subroutine EPHEM.

a. Function.

Put satellite location information into COMMON block SATEPH for use by other processing modules in the program. This information is constructed from an estimate of the DMSP orbit calculated by fitting a circular orbit to two satellite locations from the ephemeris records read in from file IESPREPFILE with each one-minute block of data frames.

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag (LOGICAL)
NEWREV	-	New readout-REV flag (LOGICAL)
EPHBLK	-	28-word ephemeris block (see 3.3.2.1)
IPHBLK	-	Integer equivalenced to EPHBLK

2. COMMON blocks.

/PIDEG/

DEG	-	Degrees to radians conversion factor
PI	-	Value of π
TWOPI	-	Value of 2π

Note: The ephemeris block is identical to the ephemeris block stored in file IESPREPFILE with the exception of word 28 in which the date (YYMMDD, integer) of the last ascending node crossing of the readout REV has been added by Subroutine RDPREP (see 2.4.3.1.)

c. Processing.

1. Remove the necessary information from the ephemeris block, converting degrees to radians and nautical miles to kilometers (1nm = 1.852km).

2. Use the date stored in word 28 of the ephemeris block to calculate the correct date. If this date is incompatible with the day-of-year in the rest of the ephemeris block, generate an error message, increment the bad ephemeris counter (NBAD), do one of the following, and return to Subroutine INPUT:

   (a) If this is the tenth bad ephemeris block in the current readout REV, write out another error message, set the return status to -2 (which alerts Subroutine INPUT to abandon this REV), disable the PRINTAWAY processor, and set the console error message to alert the operator that something has gone wrong (Subroutine PRNTON).

   (b) Otherwise, set the return status to -1, which alerts Subroutine INPUT to drop the current one minute data block and try the next one.

3. Check the times in the ephemeris block. If the difference in the two times is more than 2 seconds from the expected 60-second difference, write out an error message, jump to the error section described in step 2, and return to Subroutine INPUT.

4. Check the altitudes in the ephemeris block. If the average altitude is more than 75 km away from the nominal altitude of 825 km, write out an error message, jump to the error section described in step 2, and return to Subroutine INPUT. If the average altitude is good, also calculate the altitude change per one-second step (DSALT) and the radius of the orbit (A).

5. Calculate the nominal inclination angle for a sun-synchronous, circular orbit of radius A from:

$$i = \cos^{-1}\left[-9.892 \times 10^{-2}\left(\frac{A}{R_e}\right)^{3.5}\right]$$

where: i (DMSPI) is the orbital inclination,
R_e is the radius of the earth.

The angular (OMEGA) and orbital (VSC) velocities for this orbit are calculated from:

$$\Omega = \frac{1}{\sqrt{\left(\frac{A^3}{3.986013 \times 10^5}\right)}} \quad rad / sec$$

and

$$v_s = A\Omega \times 10^3 \quad m/sec$$

6. Determine which of the latitude/longitude pairs in the ephemeris block to use in the orbit calculation. This will always be the first pair unless the first location is too close to the equator (this can cause computational stability problems). In this case, the location farther from the equator is used.

7. Calculate the longitude (in a non-rotating frame) of the last ascending node in the current orbit (ANLON) from:

$$\theta'_a = \theta - \cos^{-1}\left(\frac{\phi_0}{\lambda}\right)$$

where: ϕ_0 (PHI0) is the angular distance along the orbit from the last ascending node to the location used in the calculation,

λ, θ are the latitude and longitude.

Calculate the inclination of the orbit (ORBINC) (this time from the observed satellite locations) from:

$$i = \cos^{-1}\left[\frac{\cos\lambda - \cos\phi_0 \cos(\theta - \theta'_a)}{\sin\phi_0 \sin(\theta - \theta'_a)}\right]$$

8. If the orbital inclination calculated from the ephemeris locations (ORBINC) is more than 5° different from the nominal inclination calculated from the altitude (DMSPI), then:

   (a) If the location used to calculate ORBINC is within 0.04 radians (~2.3°) of the equator, use the nominal inclination, as the inclination calculated from the locations may well be in error for a location close to the equator.

   (b) Otherwise, write out an error message, jump to the error section describe in step 2, and return to Subroutine INPUT.

9. Convert the ascending node longitude to the earth-fixed rotating frame, reset variable PHI0 to the value of PHI1 if the second set of locations was used in the orbit calculations, and calculate the time correction parameter used in calculating locations from the orbital parameters (T0), which is the negative of the time since the last ascending node.

10. Calculate the locations required for the arrays in COMMON block SATEPH use by the CKLPRC module. These eight locations are spaced 10 seconds apart in time, with the first time being set to

the last time previous to the current time with seconds ending in 5, i.e., if the current time (HHMMSS) is 134401, the first time would be 134355. The time (seconds since midnight) and date (day-of-year) are loaded into arrays ITCSL and IDAY, and the satellite latitude and longitude (from Subroutine LATLON), Apex latitude, longitude, and local time (from Subroutine APXTAB), and altitude for these times are loaded in the first eight locations in arrays GLAT, GLON, APXLAT, APXLON, APXLT, and ALTSC.

11. Calculate the locations used in building, and as inputs to, EDRs in the OUTPUT module. These six locations are spaced 20 seconds apart in time, with the first time being set to the last time previous to the current time with second equal to 00, i.e. if the current time is 081456, the first time would be 081400. The time (seconds since midnight) and date (YYMMDD) are loaded into arrays ITEDR and IYMD, and the same location parameters described in step 10 are loaded in the last six locations of arrays GLAT, GLON, APXLAT, APXLON, APXLT, and ALTSC.

12. Reset the time correction parameter (T0) and the orbital distance from the last ascending node (PHI0) to their values corresponding to the first time in ITCSL.

13. Set the return status to 0 and return to Subroutine INPUT.

d. Outputs.

1. Argument list.

ISTAT	-	Return status 0: No errors -1: Bad ephemeris record, skip this data block -2: Tenth bad ephemeris record, drop this REV

2. COMMON blocks.

/SATEPH/

ITCKL	-	Times for CKLPRC module (seconds since midnight)
IDAY	-	Dates for CKLPRC module (day-of-year)
ITEDR	-	Time for OUTPUT module (seconds since midnight)
IYMD	-	Dates for OUTPUT module (YYMMDD)
GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)
APXLAT	-	Apex latitude (deg)
APXLON	-	Apex longitude (deg)
APXLT	-	Apex local time (hours)
ALTSC	-	Satellite altitude (km)
VSC	-	Satellite orbital velocity (m/sec)
PHI0	-	Angle in orbit at time ITCKL(1) (rad)
OMEGA	-	Satellite angular velocity (rad/sec)
ORBINC	-	Inclination of satellite orbit (rad)
ANLON	-	Last ascending node longitude (rad)
T0	-	Time parameter (negative seconds to last ascending node)

e. Associated subroutines.

1. Subroutine TIMCON. Convert between date/time and IES minutes (see 2.4.8.12).

2. Subroutine LATLON. Calculate the geographic latitude and longitude of the satellite for an input time (see 2.4.8.7).

3. Subroutine APXTAB. Calculate the Apex latitude and longitude for an input geographic latitude and longitude (see 2.4.3.2.1).

4. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

f. Interfaces.

1. Called from Subroutine INPUT.

2. Uses COMMON blocks PIDEG and SATEPH.

2.4.3.2.1 Subroutine APXTAB.

a. Function.

Use a look-up table from file IESAPEXTABLE to convert an input geographic latitude and longitude to modified Apex latitude and longitude valid for an altitude of 840 km. See reference 1.2.3o for a description of the modified Apex coordinate system. In this document, these coordinates will be simply referred to as Apex coordinates.

b. Inputs.

1. Argument list.

GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)

2. COMMON blocks.

/APXLUT/

TABLE	-	Apex coordinate look-up table

c. Processing.

1. Convert the geographic latitude and longitude to offset-dipole latitude and longitude (Subroutine OFFSET).

2. Select a 4x4-point subgrid (APXGRD) from the look-up table centered on the offset-dipole coordinates.

3. Check the selected subgrid Apex longitude values for the discontinuity at the longitude "wrap-around" point (359.999...° to 0°). If the discontinuity falls within the subgrid, bias all longitudes

less than 180° by 360° to avoid interpolation problems.

4. Calculate the distance (in degrees) in latitude (GRLAT) and longitude (GRLON) from a reference grid point for interpolation.

5. Use a polynomial interpolation routine to interpolate in both the Apex latitude and Apex longitude tables to the offset-dipole coordinates.

6. Correct the Apex longitude calculated if the biasing introduced in step 3 leads to a longitude > 360°.

7. Return to Subroutine EPHEM.

d. Outputs.

1. Argument list.

APXLAT	-	Apex latitude (deg)
APXLON	-	Apex longitude (deg)

e. Associated subroutines.

1. Subroutine OFFSET. Convert geographic latitude and longitude to offset-dipole latitude and longitude (see 2.4.3.2.2).

2. Subroutine INTERP. Interpolate within a vector table using a polynomial fit to the input table (see 2.4.3.2.3).

f. Interfaces.

1. Called from Subroutine EPHEM.

2. Uses COMMON block APXLUT.

2.4.3.2.2 Subroutine OFFSET.

a. Function.

Convert between geographic coordinates and offset dipole coordinates.

b. Inputs.

1. Argument list.

RLAT1	-	Latitude to convert from (rad)
RLON1	-	Longitude to convert from (rad)
MODE	-	Conversion direction 1: To offset dipole coordinates 2: To geographic coordinates

2. COMMON blocks.

/APXLUT/

ALT	-	Altitude of Apex look-up table (km)
XO,YO,ZO	-	Location of offset dipole center
UO	-	Offset transformation matrix

c. Processing.

1. On the first call to this routine, set up all offset constants.

2. Convert the input latitude and longitude to Cartesian coordinates.

3. If converting from offset to geographic, then:

   (a) Correct for altitude for the dipole offset.
   (b) Correct for the dipole offset.
   (c) Rotate from the offset dipole orientation to geographic orientation.

4. If converting from geographic to offset, then:

   (a) Rotate to the dipole orientation and subtract the offsets.

5. Convert the Cartesian coordinates back into latitude and longitude.

6. Return to Subroutine APXTAB.

d. Outputs.

1. Argument list.

RLAT2	-	Converted latitude (rad)
RLON2	-	Converted longitude (rad)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine APXTAB.

2. Uses COMMON blocks APXLUT, PIDEG.

2.4.3.2.3 Subroutine INTERP.

a. Function.

Interpolate in a vector list using a polynomial interpolation technique.

b. Inputs.

1. Argument list

XI	-	Value to interpolate to
NP1	-	Number of data points in the vector list
Y	-	Data vector in which to interpolate
LOG	-	Log fit switch (LOGICAL): .TRUE. - Interpolate in log10 of Y .FALSE. - Do not use logs

c. Processing.

1. Convert the inputs to double precision, and, if requested, to log10.

2. Calculate the coefficients of an interpolation polynomial (Newton form).

3. Use the polynomial to interpolate to XI.

4. Convert the interpolated value to single precision, and, if the interpolation was done in log10 of Y, anti-log the interpolated value.

5. Return to Subroutine APXTAB.

d. Outputs.

1. Argument list.

YI	-	Interpolated value

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine APXTAB.

2. Uses no COMMON blocks.

2.4.3.3 Subroutine CHEKIT.

a. Function.

Determine the validity of the data contained in the current 1 second data frame.

b. Inputs.

1. Argument list.

ICKOLS	-	Raw data values from current frame
INTIME	-	Input data frame time
ISSDGP	-	Diagnostic print switch

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

A maximum of five consecutive invalid frames is allowed before a general reset is issued for the instrument processors. Zero data frames are passed for invalid data frames or to fill forward time jumps. After five consecutive invalid or missing frames, discontinue sending zero frames and wait for the next good data frame. If there is a backward time jump, the termination flag (THEEND) is set to .TRUE. to end the run.

1. If this is the first data frame of a readout, initialize flags and counters.

2. Test various indicators of a bad frame.

   (a) Invalid cycle counter (first data word).
   (b) RAM operation or error, Test version, Dump in progress, or Checksum error flags are set in data.
   (c) This is a duplicate data frame.
   (d) Time discontinuity.

3. Keep track of number of consecutive bad frames and set flags to take the proper action for the various situations encountered.

d. Outputs.

1. Argument list.

GRESET	-	General reset flag (LOGICAL)
IFTIME	-	Data frame time to pass to processors
IQFRME	-	Data frame quality indicator -2: Time discontinuity -1: Duplicate data frame 0: Invalid data 1: Valid data
ITJUMP	-	Difference in seconds between current data frame time and previous data frame time
THEEND	-	Termination flag (LOGICAL)

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INPUT.

2. Uses no COMMON blocks.

2.4.3.4 Subroutine DECODE.

a. Function.

Extract raw data from the SSIES sensor data block, convert the raw data to engineering units, and store the converted data in COMMON block DFRAME for use by the instrument processors.

b. Inputs.

1. Argument list.

IFQUAL	-	Current data frame quality
IFTIME	-	Current data frame time
INOLS	-	Raw data values from current frame
ISSDGP	-	Diagnostic print switch

2. COMMON blocks.

/DFCON/

FCONV	-	Raw data word conversion constants

3. Files.

None.

c. Processing.

1. Store the data frame time and data validity code.

2. If this is a valid data frame:

a. Store indicators and flags which appear in all data cycles.

b. Determine the data cycle (1 or 2) and store the flags and indicators unique to the cycle type.

c. Convert and store the raw data for the data cycle type to engineering units.

d. If the data are from an IES2 or subsequent instrument configuration, invoke Subroutine DSMSTT to process the DSM subcom register data.

e. Convert and store the raw data which appear in all cycles to engineering units.

f. Jump to step 4.

3. If this is an invalid data frame, increment and check the cycle counter for rollover.

4. If the diagnostic flag is set to 4, invoke Subroutine DECDMP to write the frame data to the standard print file.

5. Return to Subroutine INPUT.

d. Outputs.

1. Argument list.

LCCNTR	-	Last frame cycle counter

2. COMMON blocks.

/DFRAME/

IDTIME	-	Data frame time (seconds past midnight)
IQFRME	-	Data frame quality ($\leq$0: bad, 1: good)
IDWORD	-	SSIES ID word
ICYCLE	-	Cycle type (1 or 2)
ICCNTR	-	Sweep program second counter (1-128 or 1-1024)
ICALIB	-	Calibration mode flag
MODEP	-	EP sensor mode: 0=A, 1=B, 2=BS, 3=C, 4=D, 5=DS, 6=E.
IDCONF	-	Configuration ID
IELRPA	-	EP or RPA current data in telemetry: 0: RPA, 1: EP, 2: Both
MONOLS	-	OLS command monitor (cycle 1)
MONDSM	-	DSM command monitor (cycle 2)
EMTMP	-	Electrometer temperature (cycle 1)
ELMP	-	MP electron analysis Electron temperature (deg K)(cycle 1) Electron density (el/cm^3)(cycle 2)
HIMP	-	MP light ion analysis Light ion temperature (deg K)(cycle 1) Light ion density (ion/cm^3)(cycle 2)
OIMP	-	MP oxygen ion analysis Oxygen ion temperature (deg K)(cycle 1) Oxygen ion density (ion/cm^3)(cycle 2)

DMLL	-	DM log level A (cycle 1) DM log level B (cycle 2)
RELWB1	-	SM Range data (volts)
WB2RNG	-	FIBA Wideband 2 Range data (volts)
VPEL	-	MP vehicle potential from EP (volts)
VPION	-	MP vehicle potential from RPA (volts)
DMSIG	-	DM signal level monitor
ELCTMP	-	DM electronics temperature
SENTMP	-	DM sensor temperature
TMPMON	-	Electronics temperature ADC
VAPER	-	VBias + VipSet (volts)
VIPMON	-	Vip monitor
VIPSET	-	Vip setting (volts)
VBIAS	-	VBias from Vaper - VipSet (volts)
CURMON	-	MP input current (amps)
VSP	-	MP ram velocity (m/s)
IEPFLG	-	MP flags for EP analysis
INPFLG	-	MP flags for RPA analysis
THERMS	-	Temperature from thermistors: RPA (cycle 1) EP (cycle 2)
IESTYP	-	IES instrument ID code: 1=IES, 2=IES2, 3=IES2A, 4=IES3
NSMFIL	-	Number of SM filters
SMV	-	SM electrometer/amplifier (volts)
EPC	-	EP current (amps)
RPAC	-	RPA current (amps)
DMOFF	-	DM offset (volts)
SMFLT	-	SM filters (volts)
DMFILT	-	DM filters (volts)

3. Files.

None.

e. Associated subroutines.

1. Subroutine DSMSTT. Determine DM, SM, and SENPOT settings from the DSM subcom words.

2. Subroutine DECDMP. Write the decoded data frame to the standard print file.

f. Interfaces.

1. Called from Subroutine INPUT.

2. Uses COMMON blocks DFRAME and DFCON.

2.4.3.4.1 Subroutine DSMSTT.

a. Function.

Accumulate bit settings from the DSM subcom over its 16 second period.

b. Inputs.

1. Argument list.

ICCNTR	-	Cycle counter
ISUBCM	-	DSM subcom words

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Accumulate the DSM subcom register bits over the 16 second subcom period (even cycles only), zero filling and flagging register bits corresponding to missing frames.

2. When the subcom cycle has completed, set the completion/error flag, and use the register values to index to the corresponding status or command values.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/RGSTAT/

IDMSTA	-	Driftmeter mode status
IWGSTA	-	Wiggle level status
IRPSTA	-	Repeller level status
IVBSTA	-	VBias/SENPOT status
IW1STA	-	WIBAN1 Range status
IW2STA	-	WIBAN2 Range status
IRFSTA	-	SENPOT relay flag status
ISTSTA	-	Subcom processing status

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine DECODE.

2. Uses COMMON block RGSTAT.

2.4.3.5 Subroutine HSKPNG.

a. Function.

Extract and keep track of sensor housekeeping data.

b. Inputs.

1. Argument list.

NEWSAT	-	Data from different satellite flag

2. COMMON blocks.

/DFRAME/

ICYCLE	-	Data cycle type (1 or 2)
IQFRME	-	Data frame quality ($\leq$0: bad, 1: good)
IESTYP	-	IES instrument ID code: 1=IES, 2=IES2, 3=IES2A, 4=IES3
DMSIG	-	DM signal level monitor
ELCTMP	-	DM electronics temperature
EMTMP	-	Electrometer temperature (cycle 1)
TMPMON	-	Electronics temperature ADC

3. Files.

None.

c. Processing.

1. Retrieve parameters of interest from the decoded data frame.

2. Calculate averages of the following parameters for each satellite.

(a) DM offset voltage zero.
(b) Electrometer temperature.
(c) DSM electronics temperature.
(d) ADC temperature.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/SCSOH/

AADCTP	-	Average ADC temperature
ADCTMP	-	ADC temperature
ADMV0	-	Average DM signal level monitor
ADSMTP	-	Average DSM equipment temperature
AELTMP	-	Average electrometer temperature
DMV0	-	DM signal level monitor
DSMTMP	-	DSM equipment temperature
ELTMP	-	Electrometer temperature

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine INPUT.

2. Uses COMMON blocks DFRAME, DMCON, and SCSOH.

2.4.4 Subroutine PROCES.

a. Function.

This subroutine acts as the driver for the various SSIES data processing modules and the QC module.

b. Inputs.

1. Argument list.

THEEND	-	End-of-data flag

2. COMMON blocks.

/DFRAME/

IELRPA	-	EP or RPA data in telemetry: 0: RPA, 1: EP, 2: Both

/PROCON/

IDORP	-	RPAPRC module processing control flag
IDOEP	-	EPPRC module processing control flag
IDODM	-	DMPRC module processing control flag
IDOSM	-	SMPRC module processing control flag
IDOMP	-	MPPRC module processing control flag
IDOCK	-	CKLPRC module processing control flag
IDOQC	-	QCPRC module processing control flag
ISWCKL	-	Data source for C_kL analysis
IRPDGP	-	RPAPRC module diagnostic output flag
IEPDGP	-	EPPRC module diagnostic output flag
IDMDGP	-	DMPRC module diagnostic output flag
ISMDGP	-	SMPRC module diagnostic output flag
IMPDGP	-	MPPRC module diagnostic output flag
ICSDGP	-	CKLPRC module diagnostic output flag
IQCDGP	-	QCPRC module diagnostic output flag
ISSDGP	-	Processing control diagnostic print flag
NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag

c. Processing.

1. If the shell diagnostic flag is set, indicate that Subroutine PROCES was entered.

2. Invoke each of the data processing modules for which the corresponding processing flag is set to a value > 0.

   Note: For IES2 and subsequent instrument configurations, the EP or RPA processing modules will not be invoked unless the IELRPA flag indicates the necessary EP or RPA data exist in the frame.

3. If the shell diagnostic flag is set, indicate that Subroutine PROCES was left.

4. Return to the main routine.

d. Outputs.

   None.

e. Associated subroutines.

1. Subroutine SMPRC. Process data from the Scintillation Meter sensor (see 2.4.4.1).

2. Subroutine EPPRC. Process data from the Electron Probe sensor (see 2.4.4.2).

3. Subroutine RPAPRC. Process data from the Retarding Potential Analyzer sensor (see 2.4.4.3).

4. Subroutine DMPRC. Process data from the Drift Meter sensor (see 2.4.4.4).

5. Subroutine MPPRC. Process sweep analyses from the on-board microprocessor (see 2.4.4.)

6. Subroutine CKLPRC. Calculate scintillation parameters from the SM or EP sensor density data (see 2.4.4.6).

7. Subroutine QCPRC. Perform data quality control and build the output record for the processing statistics file (see 2.4.4.7).

f. Interfaces.

1. Called from the main routine.

2. Uses COMMON blocks DFRAME and PROCON.

2.4.4.1 Subroutine SMPRC.

a. Function.

Control processing of the SM data processing module which converts raw sensor data from the SSIES Scintillation Meter to estimates of ion density, N_i, and ΔN_i power spectral density (PSD). Figure 8 is a hierarchy chart for the SM data processing module. A description of the processing algorithm can be found in Appendix B.4.

SMPRC:	process scintillation meter (SM) data
. RANGE:	interpret electrometer/SM-wideband range setting value
. . SMCMD:	interpret SM command value
. PRNTON:	set PRINTAWAY status for conclusion of processing
. ELAMP:	convert electrometer/SM-wideband values to ion density
. . COPY:	initialize array or transfer values between arrays
. . FND511:	search for range change flag in SM data sequence
. . PRMRNG:	check environment parameters against allowed ranges
. COPY:	initialize array or transfer values between arrays
. FILTER:	convert SM filter band values to power spectral density values
. SMDIAG:	report SM processing diagnostics
. . OPNDOF:	initialize diagnostic output file
. . . BITON:	test for bit setting
. . . FILERR:	determine and report file error status
. . . PRNTON:	set PRINTAWAY status for conclusion of processing
. . FILERR:	determine and report file error status

Figure 8 SMPRC module hierarchy chart

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite switch
NEWREV	-	New readout REV switch
GRESET	-	General reset (LOGICAL)
THEEND	-	End-of-data flag
IDOSM	-	SM processing control switch 0: Do not process SM data 1: Process only density data 2: Process only filter data 3: Process all data
IDIAG	-	SM diagnostic print switch 0: Produce no diagnostic print 1: Produce a diagnostic print file

2. COMMON blocks.

/DFRAME/

IDTIME	-	Data frame time (seconds after midnight, GMT)
IESTYP	-	IES instrument ID code: 1=IES, 2=IES2, 3=IES2A, 4=IES3
IQFRME	-	Data frame quality (≤0: bad, 1: good)
SMV	-	One second (24 samples of SM electrometer data (volts)
SMFLT	-	One second of SM filter data (volts)
RELWB1	-	SM range data (volts)
MONDSM	-	DSM command monitor

/RPAANL/

RPAUR	-	Ram drift velocity (meters/sec)

/SATEPH/

VSC	-	Satellite velocity (meters/sec)

/SMCON/

SMAEFF	-	SM effective area (cm^2)
SMEQ	-	SM electrometer amplifier gain
SMAG	-	SM difference amplifier gain
SMAOFF	-	SM difference amplifier offset (volts)
SMRFIT	-	Range data fit envelope (volts)
VTABLE	-	Range data tables (volts)
SMFG	-	SM filter amplifier gains
SMEOFF	-	SM filter amplifier offsets (volts)
SMFBW	-	SM-filter amplifier bandwidths (Hertz)
NSMFLT	-	Number of SM filters

ISMUR - Ram ion velocity data-use flag
0: Do not use RPA ram drift velocity
1: Use RPA ram drift velocity

c. Processing.

1. Reset processing flags for a new satellite, new REV, or general reset. If no more data, return to Subroutine PROCES.

2. Determine the range settings on the SM electrometer and wide-band amplifiers (Subroutine RANGE). If an error was encountered alert the user, turn on the print (Subroutine PRNTON), and return. If the range data has been significantly different than expected (Subroutine RANGE), turn on the print (Subroutine PRNTON).

3. Calculate the total velocity of the ions with respect to the sensor ($V_{sc} + u_r$). If the u_r use flag is zero, use V_{sc} only.

4. If the filter data are to be processed, convert the voltages from the wideband amplifier to PSD estimates (Subroutine FILTER). If an error was encountered, alert the user and turn on the print (Subroutine PRNTON). If not, set the SM processing flag to its current value plus 2.

5. If requested, generate diagnostic output (Subroutine SMDIAG).

6. Return to Subroutine PROCES.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

SMIDEN	-	24 ion densities (ion/cm^3)
SMADEN	-	Average ion density (ion/cm^3)
SMFPDS	-	PSD estimates ((ion/cm^3)^2/Hz)
ISMEF	-	SM processing error flag
ISMAR	-	SM analysis results flag

e. Associated subroutines.

1. Subroutine RANGE. Process the SM range data (see 2.4.4.1.1).

2. Subroutine ELAMP. Process SM electrometer/amplifier data (see 2.4.4.1.3).

3. Subroutine FILTER. Process SM filter data (see 2.4.4.1.5).

4. Subroutine SMDIAG. Generate diagnostic output (see 2.4.4.1.6).

5. Subroutine PRNTON. Set a flag to disable the runstream PRINTAWAY (see 2.4.8.11).

6. Function BITON. Determine if a particular bit is set in a word (see 2.4.8.2).

f. Interfaces.

1. Called by Subroutine PROCESS.

2. Uses COMMON blocks DFRAME, LUDIAG, RPAANL, SATEPH, SMANL, and SMCON.

2.4.4.1.1 Subroutine RANGE.

a. Function.

Convert the SM range data word (RELWB1) to the ranges on the SM electrometer and wideband amplifiers.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV reset flag (logical)
IQFRME	-	Data frame quality flag
RELWB1	-	Range data (volts)
VTABLE	-	Range data table (volts)
MONDSM	-	DSM command monitor

c. Processing.

1. Reset last settings as necessary, and return to Subroutine SMPRC if the current data frame is bad.

2. Determine if the wideband amplifier has been commanded to a new operating mode (Subroutine SMCMD).

3. Determine the wideband amplifier range setting by searching the range table.

4. Determine the electrometer range setting by searching the section of the range table which corresponds to the wideband amplifier setting.

(a) If a fit is found, set the electrometer and wideband amplifier ranges.

(b) If not, set the electrometer range parameter to zero and increment the error parameter.

(c) If the fit error counter exceeds 100, alert the user and set the error flag to 2.

5. Return to Subroutine SMPRC.

d. Outputs.

1. Argument list.

IELR	-	Electrometer range bin (1-5)
IWBR	-	Wideband amplifier range bin (1-5)
NEWRNG	-	New range flag (either amplifier)
NEWCMR	-	New SM command lag (range changes only)

IERR - Error flag
0: No errors
1: No range fit found
2: Range data too far from nominal values

e. Associated subroutines.

1. Subroutine SMCMD. Process an SM command (see 2.4.4.1.2).

f. Interfaces.

1. Called by Subroutine SMPRC.

2. Uses no COMMON blocks.

2.4.4.1.2 Subroutine SMCMD.

a. Function.

Determine if the SM wideband amplifier has been commanded into a new operating mode.

NOTE: SMCMD processing is specific to **IES, IES2 and IES2A** only. When IES3 comes on line, SMCMD may need to be modified to handle IES3 DSM commands.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV reset flag (logical)
IESTYP	-	IES instrument ID code: 1=IES, 2=IES2, 3=IES2A, 4=IES3
MONDSM	-	DSM command monitor

2. COMMON blocks.

/RGSTAT/

ISTSTA	-	Subcom processing status
IW1STA	-	WIBAN1 Range status

c. Processing.

1. If RESET is .TRUE., set the last command value to -1.

2. If the DSM command matches the previous one, set the new command flag (NEWCMR) to .FALSE. and return to Subroutine RANGE.

3. If this is a new command and:

(a) this an IES data configuration, check the command for validity.

(b) this is an IES2 or IES2A data configuration, check the command for validity. If the command is not valid, try to obtain the command from the subcom status.

4. If the command was valid, set NEWCMR to .TRUE.. Otherwise, set NEWCMR to .FALSE.

5. Return to Subroutine RANGE.

d. Outputs.

1. Argument list.

NEWCMR	-	New command flag (logical): .TRUE.: The range was reset .FALSE.: The range was not reset

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine RANGE.

2. Uses no COMMON blocks.

2.4.4.1.3 Subroutine ELAMP.

a. Function.

Convert 24 EL/AMP voltage samples from the SM sensor to estimates of ion density. NOTE: This is a very complicated and, consequently, very touchy, subroutine. DO NOT make any changes to this routine until you completely understand the description of the processing algorithm for the SM EL/AMP data in Appendix B.4 and in reference 1.2.3d, and completely understand the data flow in this routine.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV reset flag
IQFRME	-	Data frame quality flag
SMV	-	24 SM EL/AMP voltages
VTCON	-	Constant equal to SMAEFF $\times$ VTOT $\times$ 10^6
SMEG	-	Electrometer (EL) gain
SMAG	-	Difference amplifier (AMP) gain
SMAOFF	-	Difference amplifier offset (volts)
IRDELR	-	EL range bin (1-5) from range data
ETABLE	-	Imbedded range flag data table (volts)

c. Processing.

1. If required, reset internal control parameters.

2. If the current data frame is bad, place processing into a wait status, reset internal control parameters, and return to Subroutine SMPRC.

3. If in a wait status, check for a valid 5.11 EL-to-AMP flag in the current set of voltages (Function FND511). If none is found, remain in wait status. If the last EL range setting from the range data word is valid (IRDELR), use it to set the EL range gain. If not, remain in wait status.

4. If in wait status, return to Subroutine SMPRC.

5. Initialize processing-loop control parameters.

6. Start main processing loop.

7. Set variables LOW and HIGH to flag if the current voltage is below 0.32 or above 5.10 volts, respectively.

8. If the voltage is neither low nor high, or if the low density flag is set (LOWDEN), calculate an ion density from the current voltage value as follows:

    (a) If in AMP mode, then

        (1) If a valid base density from the last EL measurement is not available, go into wait status and jump to step 13.

        (2) Calculate a delta-current from the latest voltage, use it to calculate a delta-density, and calculate the total ion density by adding the delta-density to the base density from the last EL measurement.

    (b) If in EL mode, calculate the current from the latest voltage and calculate the total ion density from this current.

    (c) In either mode, if the calculated density is within allowed limits, load the density in the output density array and add it to the density sum to be used in calculating the average density for the current one-second period. If in EL mode, save the density and voltage for future processing of AMP data. Reset the low density flag to .FALSE..

9. If the voltage is high, it may be an EL-to-AMP flag. Process as follows:

    (a) If in AMP mode, and there is no break between the last good data frame and the current one, something is wrong. Place processing in a wait status and jump to step 13.

    (b) If in EL mode, and if the current voltage is not in word 1, 2, or 5 of the 24-point voltage set, place processing in a wait status and jump to step 13. Otherwise, set the AMP flag to .TRUE..

10. If the voltage is low, it may be either an AMP-to-EL flag with or without an EL range change, an EL range change, or a density data point if EL is in range 1. Process as follows:

(a) Calculate a range setting (NEWELR) from the voltage using the range table for imbedded range flags (ETABLE).

(b) If in EL mode, then

(1) If the EL range setting was 1, this may be a data point rather than a flag. If the new setting is 2, and

(i) the current voltage is prior to the range data measurement time (i.e., is prior to data point #10) and the range data EL setting agrees that the setting is 2,

or

(ii) the voltage value in ELSAVE is above 0.75 volts,

this is a range change flag. If the new setting is not 2, set the low density flag and jump back to step 8. If this was neither a low density point nor a range change, place processing in a wait status.

(2) If the EL range setting was 2, then a check must be made to differentiate between a valid change to range 1 and a density data point in the range 0.30-0.32 volts. If the new range is 2 or greater than 3, place processing in a wait status. If the new range is 1 and the voltage is greater than, or equal to, 0.30 volts, this is a range change if either

(i) the current voltage is prior to the range data measurement time (i.e., is prior to data point #10) and the range data EL setting agrees that the setting is 1,

or

(ii) the voltage value in ELSAVE is above 0.75 volts.

If not, this is a density data point; set the low density flag and jump back to step 8.

(3) If the last range setting was greater than 2, and the current voltage is greater than, or equal to, 0.30, this is a density data point. Set the low density flag and jump back to step 8.

(4) If the new range setting is 5, and the range data EL range indicates the setting is below 4, place processing in wait status.

(c) If in AMP mode, then

(1) If no valid range setting was found in step 10a, and the current voltage is above the lowest voltage in the imbedded-flag look-up table, place processing in wait status.

(2) If this is an IES2 or IES2A configuration and this is the last sample, and a switch from AMP to EL mode occurred on the 16^{th} sample, place processing in wait status.

(3) If the new range is 5, then a check must be made to differentiate between a range change and an AMP-to-EL flag with no range change. If the last range setting was not 4, assume that this is really a no-change flag. If the last range setting was 4, and the current voltage is prior to the range data measurement time (i.e., is prior to data point #10) and the range data EL setting agrees that the setting is 5, this was a valid range change from 4 to 5. If not, place processing in wait

status.

(d) If any of the above checks have placed processing in wait status, jump to step 13.

(e) If the range setting is non-zero, calculate a new EL range gain and save the new range setting.

(f) Set the AMP flag to .FALSE., reset the base density and EL voltage variables, and increment the voltage-array index by 1 to skip over the next voltage (which should be a second AMP-to-EL or range change flag).

11. If this is data point 10 in IES configuration, or data point 6 in IES2 or IES2A configuration, the range setting obtained from imbedded flags should agree with the setting from the range data word. If not, alert the user, place processing in wait status, and jump to step 13.

12. Increment the voltage array index. If it is less than 25, jump back to the start of the processing loop (step 6) and process the next voltage.

13. END OF MAIN PROCESSING LOOP.

14. If not in wait status, set the starting value of the voltage-array index for the next one-second data set and calculate the average density for the present one-second data set. If in wait status, reset all control parameters.

15. If processing was suspended during this data set, increment the appropriate counter and alert the user.

16. Return to Subroutine SMPRC.

d. Outputs.

1. Argument list.

IVDELR	-	EL range bin (1-5) from imbedded flags
SMIDEN	-	24 ion density measurements(ion/cm^3)
SMADEN	-	Average ion density (ion/cm^3)

e. Associated subroutines.

None.

1. Function PRMRNG. Check the range of the density measurements calculated (see 2.4.8.10).

2. Function FND511. Find a valid 5.11 volt flag.

f. Interfaces.

1. Called from Subroutine SMPRC.

2. Uses no COMMON blocks.

2.4.4.1.4 Function FND511.

a. Function.

Determine if a 5.11-volt flag is located at word #1 or word #5 in a 24-point array of SM EL-AMP data.

b. Inputs.

1. Argument list.

AMV	-	SM EL-AMP voltages (24)
ELSAVE	-	Last good EL voltage

c. Processing.

1. Search through the voltage array for a 5.11-volt flag. Set variable I511 to the location in array SMV if one is found.

2. If a flag was found in word #1 and ELSAVE contains a valid voltage, set word #1 to the value in ELSAVE, set word #2 to 5.11, set the start-processing index to 1, and set FND511 to .TRUE.. If ELSAVE is not valid, save the last voltage in the array in ELSAVE and set FND511 to .FALSE..

3. If a flag was found in word #5, set the start-processing flag to 4 and set FND511 to .TRUE..

4. If flags were not found in words 1 or 5, save the last voltage in array SMV in ELSAVE and set FND511 to .FALSE..

5. Return to Subroutine ELAMP.

d. Outputs.

1. Function value.

FND511	-	.TRUE. if a valid EL-to-AMP flag was found

2. Argument list.

ELSAVE	-	Set to last good EL voltage in SMV
ISTART	-	Start-of-processing index (1 or 4)

e. Associated subroutines.

None.

f. Interfaces.

1. Invoked by Subroutine ELAMP.

2. Uses no COMMON blocks.

2.4.4.1.5 Subroutine FILTER.

a. Function.

Convert the SM filter bank voltages to estimates of N_i power spectral density (PSD).

b. Inputs.

1. Argument list.

SMFLT	-	SM filter bank voltages (volts)
NSMFLT	-	Number of SM filter bank voltages (normally 9 for IES; 6 for IES2 & IES2A)
VTCON	-	Constant equal to SMAEFF $\times$ VTOT $\times 10^6$
SMFG	-	SM filter amplifier system effective gain
SMFOFF	-	SM filter amplifier offsets (volts)
SMFBW	-	SM filter amplifier bandwidths (hertz)
IRDELR	-	Wideband amplifier range bin (1-5)
NOFILT	-	SM filter data use flag (LOGICAL)

c. Processing.

1. If the data are to be processed, then convert each of the filter readings for which the bandwidth is non-zero to PSD estimates as follows:

   (a) Calculate the RMS variance (RMSDI) in the current from the filter voltage.

   (b) Convert this to an RMS variance in density (RMSDNE).

   (c) Convert this to an estimate of the PSD (RMSPDS).

2. Return to Subroutine SMPRC.

d. Outputs.

1. Argument list.

SMFPDS	-	estimates of N_i power spectral density

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine SMPRC.

2. Uses no COMMON blocks.

2.4.4.1.6 Subroutine SMDIAG.

a. Function.

Construct diagnostic output files for the SM data processing module.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag
IRDELR	-	Electrometer amplifier range bin (1-5)
IWBR	-	Wideband amplifier range bin (1-5)
IDIAG	-	Diagnostic output file: 0: No output 1: Binary diagnostic output only 2: ASCII diagnostic output only 3: Both ASCII and binary output

2. COMMON blocks.

/DFRAME/

IDTIME	-	Data frame time (seconds since midnight)
IQFRME	-	Data frame quality flag
MONDSM	-	DSM command monitor
SMFLT	-	SM filter bank voltages
RELWB1	-	SM range data
SMV	-	SM EL/AMP voltages

/LUDIAG/

LUSMA	-	Logical unit for ASCII output
LUSMB	-	Logical unit for binary output

/PROCON/

IFLT	-	Satellite flight ID

/SMANL/

SMIDEN	-	Ion density (24, ion/cm^3)
SMADEN	-	Average ion density (ion/cm^3)
SMFPDS	-	Irregularity PDS estimates (9) ((ion/cm^3)^2/Hz)
ISMEF	-	SM analysis error flag
ISMAR	-	SM analysis results flag

/SMCON/

SMEG	-	Electrometer amplifier gain
SMAG	-	Difference amplifier gain
SMAOFF	-	Difference amplifier offset (volts)

SMRFIT	-	Fit criterion for range data (volts)
VTABLE	-	Range table (volts)
SMFG	-	Filter amplifier gain
SMFOFF	-	Filter offsets (volts)
SMFFRQ	-	Filter bank center frequencies (Hz)
SMFBW	-	Filter bank bandwidths (Hz)
NSMFLT	-	Number of filters in filter bank
ISMUR	-	RPA ram drift velocity use flag

c. Processing.

1. On first call to this routine, open the necessary ASCII and/or binary output files (Subroutine OPNDOF).

2. If neither file could be opened, set the diagnostic print flag to zero (disabling the diagnostic output) and return to Subroutine SMPRC.

3. If an ASCII output file is requested, then

   (a) If a RESET=.TRUE. call, output the sensor processing constants and processing control information for the SM module.

   (b) If the data frame was good, write out all data input and produced in the current invocation of the SM processor. (Note: If the frame was bad, write out a note to that effect).

4. If a binary output file is requested and if the frame was good, write the diagnostic output buffer to the output file.

5. Return to Subroutine SMPRC.

d. Outputs.

IESSMPRT. ASCII diagnostic output file (see Section 4.4.5).

IESSMDIAG. Binary diagnostic output file (see Section 4.4.5).

e. Associated subroutines.

1. Subroutine OPNDOF. Open diagnostic output files (see 2.4.8.9).

2. Subroutine FILERR. File error handling routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine SMPRC.

2. Uses COMMON blocks DFRAME, LUDIAG, PROCON, SMANL, and SMCON.

2.4.4.2 Subroutine EPPRC.

a. Function.

Control the processing of Electron Probe (EP) data. Convert currents and voltages from the EP into the environmental parameters electron density, electron temperature, and satellite potential. Figure 9 is a hierarchy chart for this module.

EPPRC:	process electron probe (EP) data
. SWPCOL:	accumulate data for a complete sweep analysis
.. COPY:	initialize array or transfer values between arrays
. EPSWP:	process EP sweep data for density, temperature, and potential
.. COPY:	initialize array or transfer values between arrays
.. EPDIAG:	report EP processing diagnostics
... OPNDOF:	initialize diagnostic output file
.... BITON:	test for bit setting
.... FILERR:	determine and report file error status
.... PRNTON:	set PRINTAWAY status for conclusion of processing
... LOGLIN:	transform tabulated values between standard and log10 form
... FILERR:	determine and report file error status
.. FITLIN:	perform linear least-squares fit for tabulated values
.. NTRP:	perform two-point interpolation
.. PRMRNG:	check environment parameters against allowed ranges
. SELNRM:	select normalization density or temperature for EP DC analysis
. EPDC:	process EP DC (dwell) data for electron density
.. PRMRNG:	check environment parameters against allowed ranges

Figure 9 EPPRC module hierarchy chart

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag
THEEND	-	End of data flag
IDOEP	-	EP data processing flag 0: Do not process ≥1: Process all data
IEPDGP	-	Diagnostic output flag 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files

2. COMMON blocks.

/DFRAME/

EPC	-	EP currents for this data frame

ICALIB	-	Calibration sweep identifier flag
ICCNTR	-	EP instrument cycle counter
IDTIME	-	Time of this data frame (seconds since midnight GMT)
IQFRME	-	Data frame quality flag
MODEP	-	EP mode flag
VBIAS	-	Instrument bias voltage

/EPCON/

EPAREA	-	EP collector area
EPCVIP	-	EP calibration sweep start test voltage
IPDTMP	-	Default electron temperature for density calculations
EPNVDN	-	EP normal down sweep start test voltage
EPNVUP	-	EP normal up sweep start test voltage

/EPANL/

IEPAR	-	EP analysis result flag

/CONSTS/

E	-	Electron charge
K	-	Boltzmann's constant
ME	-	Mass of an electron

/PIDEG/

PI	-	π

3. Files.

None

c. Processing.

1. If end of data flag (THEEND) is .TRUE., return to Subroutine PROCES.

2. If first pass (FIRST), or reset (GRESET), or new satellite (NEWSAT), or new readout (NEWREV), then reset the necessary flags and counters.

2. If these data are from Mode BS or Mode DS and the calibration flag (ICALIB) is set, terminate any ongoing sweep.

3. If the previous sweep is finished and this is a good data frame:

(a) Skip the one second of data after each calibration sweep. (Calibration sweeps are three seconds duration.) This is cycle counter 4 and 8.

(b) If conditions are right for the start of a calibration sweep, set the collection variables to initiate the sweep collection.

(c) Otherwise, if the conditions are right for the start of a normal sweep, set the collection

variables to initiate the sweep collection.

(d) Otherwise, if we are in Mode C or D and the conditions are right for the start of a DC mode 4 second up sweep, or the 28 second dwell, set the collection variables to initiate the sweep or dwell collection.

(e) Otherwise, if we are in Mode E and the conditions are right for the start of a short (2 second) sweep, set the collection variables to initiate short sweep collection.

4. If the sweep type is not a 28 second dwell then:

(a) If the sweep status is new or ongoing, invoke Subroutine SWPCOL to collect this second's worth of data for the sweep.

(b) If the sweep collection is complete, set the EP analysis results time and invoke Subroutine EPSWP to process the sweep.

(c) If the completed sweep was a DC Mode 4 second sweep, invoke Subroutine SELNRM to get the normalization density and/or electron temperature to use in DC mode density calculations.

5. Otherwise, the collection is a 28 second dwell, so invoke Subroutine EPDC to calculate the electron densities.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/EPANL/

IEPTIM - EP analysis result time

3. Files.

None.

e. Associated subroutines.

1. Subroutine SWPCOL. Collects the sweep data (See 2.4.4.2.1).

2. Subroutine EPSWP. Process an EP sweep (See 2.4.4.2.2).

3. Subroutine SELNRM. Select normalization sensor for DC mode and set normalization parameters (See 2.4.4.2.5).

4. Subroutine EPDC. Process EP DC mode data (See 2.4.4.2.6).

f. Interfaces.

1. Called from Subroutine PROCES.

2. Uses COMMON blocks DFRAME, EPCON, EPANL, CONSTS, and PIDEG.

2.4.4.2.1 Subroutine SWPCOL.

a. Function.

Collect successive data frames to build a complete EP or RPA sweep.

b. Inputs.

1. Argument list.

ICCNTR	-	Data cycle counter
SWPCUR	-	EP or RPA raw data frame currents
SSCEC	-	RPA sensor sweep coupling error correction factor
ISEORR	-	EP or RPA sweep indicator (0=EP, 9=RPA)
ISTAT	-	Sweep status
ISWTYP	-	Sweep type
ISWDIR	-	Sweep direction

2. COMMON blocks.

/SWPVLT/

SWPVLT	-	Array of 'ideal' sweep voltages.

3. Files.

None.

c. Processing.

1. If current data frame is the start of a new sweep, save the instrument type and the sweep type and direction variables, initialize the sweep collection current and voltage arrays to zero, and tag the sweep as ongoing.

2. Otherwise, if we have missed a data cycle or changed instruments since the last cycle, terminate the ongoing sweep.

3. If these data are part of an ongoing sweep, update the indices for sweep direction, sweep type, and sweep time, and store the current data into the sweep. If these data are for the RPA, convert them to linear values (from log10) and apply the sensor correction factor.

4. Update the counters and indices for the next second of data.

5. If the sweep is complete, set the sweep status to done and reconstruct the missing 25th sample from each second by simple linear interpolation.

6. Return to the invoking subroutine.

d. Outputs.

1. Argument list.

CL	-	Complete sweep current array: Log10 for EP, Linear for RPA
NVALS	-	Number of values collected in sweep
V	-	Complete sweep voltage array

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

Subroutine COPY. Copies one array to another, or a scalar into specified elements of an array. (see 2.4.8.3)

f. Interfaces.

1. Called from EPSWP and RPASWP.

2. Uses no COMMON blocks.

2.4.4.2.2 Subroutine EPSWP.

a. Function.

Process the current vs. voltage data provided by the Electron Probe. Successful analysis will result in determination of electron density, electron temperature, and satellite potential.

b. Inputs.

1. Argument list.

CL	-	Sweep current array
IEPDGP	-	Diagnostic output switch 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files
NUM	-	Number of points in sweep
V	-	Sweep voltage array
VBIAS	-	Instrument bias voltage

2. COMMON blocks.

/EPCON/

EPAREA	-	EP collector area
EPCCC1	-	Intersection point correction coefficient 1
EPCCC2	-	Intersection point correction coefficient 2
EPCCC3	-	Intersection point correction coefficient 3
EPCRNG	-	Required current range to make a sweep analysis
EPMDEL	-	Change in log current allowable near sweep start
EPNSDN	-	Acceptable noise level decrease in current
EPNSUP	-	Acceptable noise level increase in current
EPTPAR	-	Temperature calculation constant
EPTRAN	-	EP instrument transparency
IEPCOR	-	Make plasma potential estimation correction flag
IEPMIN	-	Minimum number of points allowed for sweep analysis

/CONSTS/

E	-	Electron charge
K	-	Boltzmann's constant
ME	-	Mass of an electron

/PIDEG/

PI	-	π

3. Files.

None.

c. Processing.

1. Initialize variables as necessary.

2. Find valid sweep starting point.

3. Clean up the sweep. Successive points within the sweep must be within expected noise limits of each other.

4. Locate the minimum sweep current. Test the sweep range against the criteria required to perform the analysis.

5. Fit a straight line to the saturation region currents and test them for true linearity.

6. Locate the point where the sweep curve first reaches its maximum slope by fitting a straight line to a specified number of consecutive points.

7. Locate the intersection of the lines found in steps 5 and 6.

8. If requested, apply a correction term based upon a parametric analysis of the offset of the intersection from the true point of interest specified in step 6.

9. Calculate the required environmental parameters.

d. Outputs.

1. Argument list.

CVB	-	Sweep current at the point of the sensor potential

2. COMMON blocks.

/EPANL/

EPEDEN	-	EP electron density
EPETMP	-	EP electron temperature
IEPAR	-	EP analysis result flag
NEPSG	-	Number of good sweep analyses this readout
NEPST	-	Number of sweeps attempted this readout

3. Files.

None.

e. Associated subroutines.

1. Subroutine EPDIAG. Provides diagnostic information for the user (see 2.4.4.2.3).

2. Subroutine NTRP. Performs a two-point interpolation from a table of function arguments and values (see 2.4.4.2.4).

f. Interfaces.

1. Called from Subroutine EPPRC.

2. Uses COMMON blocks EPCON, EPANL, CONSTS, and PIDEG.

2.4.4.2.3 Subroutine EPDIAG.

a. Function.

Provide diagnostic messages to the user concerning problems encountered during EP data processing. Additionally, provide a copy of the raw EP sweep for further diagnostic study at a later time.

b. Inputs.

1. Argument list.

VOLT	-	EP raw sweep voltage array
CURR	-	EP raw sweep current array
MSG	-	Diagnostic message number

IDIAG	-	EP diagnostic output flag 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files

2. COMMON blocks.

/DFRAME/

IDTIME	-	Data frame time (seconds since midnight GMT)
IQFRME	-	Data frame quality flag
MODEP	-	EP mode flag
VBIAS	-	Instrument bias voltage

/LUDIAG/

LUEPA	-	Logical file number for EP ASCII diagnostic file
LUEPB	-	Logical file number for EP binary file

/PROCON/

IFLT	-	Vehicle flight number

/EPANL/

EPEDEN	-	EP electron density
EPETMP	-	EP electron temperature
IEPTIM	-	EP sweep time (seconds since midnight GMT)
EPVPOT	-	EP sensor potential

3. Files.

None.

c. Processing.

1. The design is for a short, single line, diagnostic concerning any problem encountered in the sweep analysis so that any trends or repetitive data problems can be isolated.

2. On first call, open the requested diagnostic files. The binary file will contain the raw EP sweep and any analysis accomplished by this processor. The ASCII file will contain any diagnostic messages relating to the processor analysis.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

None.

3. Files.

Diagnostic print to files IESEPPRT and IESEPDIAG.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from EPSWP.

2. Uses COMMON blocks DFRAME, LUDIAG, PROCON, EPANL, and EPCON.

2.4.4.2.4 Subroutine NTRP.

a. Function.

Perform a two-point interpolation based upon a table of argument values and a table of corresponding function values.

b. Inputs.

1. Argument list.

N	-	Number of values in tables
X	-	Argument at which interpolated function value is desired
XTAB	-	Table of argument values
YTAB	-	Table of function values

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Locate the interpolation boundaries.

2. Perform the linear interpolation.

d. Outputs.

1. Argument list.

Y	-	Interpolated function value for specified argument value

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine EPSWP.

2. Uses no COMMON blocks.

2.4.4.2.5 Subroutine SELNRM.

a. Function.

Select the normalization density and/or electron temperature to use in DC mode density calculations.

b. Inputs.

1. Argument list.

CVB	-	Current value at sensor potential

2. COMMON blocks.

/EPCON/

EPDTMP	-	Default electron temperature
NEPDM	-	Normalize to DM density flag
NEPRP	-	Normalize to RPA density flag
NEPSM	-	Normalize to SM density flag

/EPANL/

EPEDEN	-	EP electron density
EPETMP	-	EP electron temperature
IEPAR	-	EP analysis result flag

/RPAANL/

IRPAR	-	RPA analysis result flag
RPEDEN	-	RPA estimated electron density

/SMANL/

ISMAR	-	SM analysis result flag
SMADEN	-	SM estimated electron density

/DMANL/

DMIDEN	-	DM estimated electron density
IDMAR	-	DM analysis result flag

3. Files.

None.

c. Processing.

1. Examine the analysis results flags for the possible normalizing density values in the established priority order. The first instrument which was selected in IESCNTRLFILE as a possible normalizer will be used if an analysis is available for that particular instrument. If a normalizer flag could not be found set the NORMLZ flag to false.

2. Select a temperature to use in un-normalized density calculations. Use the EP sweep analysis temperature if available. If not available use the default temperature from IESCNTRLFILE.

d. Outputs.

1. Argument list.

FNORM	-	Normalization factor
NORMLZ	-	Perform normalization flag
TE	-	Electron temperature for un-normalized density calculations

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine EPPRC.

2. Uses COMMON blocks EPCON, EPANL, RPAANL, SMANL, and DMANL.

2.4.4.2.6 Subroutine EPDC.

a. Function.

Calculate electron densities from the DC modes (C and D data).

b. Inputs.

1. Argument list.

EPC	-	EP currents for this data frame
FNORM	-	Density normalization factor
TE	-	Electron temperature for un-normalized density calculations
CONST	-	Instrument parameter calculation constant
NORMLZ	-	Normalize the calculated densities flag

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

Loop through the 24 current values performing the necessary calculations:

(a) If we are to normalize, multiply the current by the normalization factor.

(b) If we are not normalizing, calculate the density based upon the instrument parameters and an established electron temperature.

(c) Test the density value for appropriate range and calculate an average density. Average current is calculated for diagnostic purposes.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/EPANL/

EPADEN	-	Average electron density for this data frame
EPEDEN	-	Densities for each current value
IEPAR	-	EP analysis result flag

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from EPPRC.

2. Uses COMMON block EPANL.

2.4.4.3 Subroutine RPAPRC.

a. Function.

Control the processing of Retarding Potential Analyzer (RPA) sweeps. Convert currents and voltages from the RPA into environmental parameters; ion density, ram direction drift velocity, and sensor potential. Figure 10 is a hierarchy chart for this module.

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag
THEEND	-	End of data flag
IDORP	-	Process RPA data 0: Do not process ≥1: Process all sweeps
IRPDGP	-	Diagnostic output flag 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files

2. COMMON blocks.

/DFRAME/

ICALIB	-	Calibration mode indicator flag
IDTIME	-	Input data frame time
IQFRME	-	Input data frame quality
RPAC	-	RPA currents for this data frame
VBIAS	-	Bias voltage
VAPER	-	Aperture voltage (VBIAS + VIP)

RPAPRC:	process retarding potential analyzer (RPA) data
. SWPCOL:	accumulate data for a complete sweep analysis
. RPAALG:	process RPA sweep data for density, temperature, and potential
.. COPY:	initialize array or transfer values between arrays
.. LOGLIN:	transform tabulated values between standard and log10 form
.. CHKSWP:	pre-process and validate RPA sweep data
... RPDIAG:	report RPA processing diagnostics
.... OPNDOF:	initialize diagnostic output file
..... BITON:	test for bit setting
..... FILERR:	determine and report file error status
..... PRNTON:	set PRINTAWAY status for conclusion of processing
.... FILERR:	determine and report file error status
... MEANSD:	calculate mean and standard deviation for list of values
.. XPOS:	determine voltages for specified sweep currents
... COPY:	initialize array or transfer values between arrays
.. RPDIAG:	report RPA processing diagnostics
.. SSLOPE:	find local minimum in slope of sweep profile
... COPY:	initialize array or transfer values between arrays
... FITLIN:	perform linear least-squares fit for tabulated values
.. PLASMA:	determine environmental parameters from engineering data
... ERF:	error function for normal (Gaussian) distribution
.. PRMRNG:	check environment parameters against allowed ranges
.. RPASWP:	calculate instrument response to specified plasma environment
... ERF:	error function for normal (Gaussian) distribution
.. COMPAR:	determine variance between measured and theoretical sweeps
.. ANLSAV:	store RPA sweep solution
.. FITLIN:	perform linear least-squares fit for tabulated values

Figure 10 RPAPRC module hierarchy chart

/RPACON/

RPCPLG	-	Sweep current coupling error correction
RPDSPS	-	Default satellite potential, sunlight
RPSVDN	-	Down sweep start voltage test
RPSVUP	-	Up sweep start voltage test

/SATEPH/

APXLAT	-	Magnetic latitude of spacecraft
ITCSL	-	Expanded ephemeris time information

/RPAANL/

IRPAR	-	RPA analysis results flag

/EPANL/

EPVPOT	-	EP estimated sensor potential
IEPAR	-	EP analysis results flag

3. Files.

   None.

c. Processing

   1. If end-of-data flag (THEEND) is set, return to Subroutine PROCES.

   2. If first pass, reset condition (GRESET), new satellite (NEWSAT), or new readout (NEWREV), reset the necessary flags and counters.

   3. If this is a good data frame and the previous sweep has ended, test the cycle counter to determine if this cycle is the start point for a new sweep. If so, establish the proper sweep direction and set sweep status to new.

   4. If this cycle is part of a new or ongoing sweep:

      (a) Invoke SWPCOL to gather the RPA sweep data.

      (b) If the sweep collection is complete:

         (1) Set the analysis results time.

         (2) Get magnetic latitude from ephemeris block.

         (3) Set default satellite potential.

         (4) Invoke Subroutine RPAALG to process the sweep.

   (5) Return to Subroutine PROCES.

d. Outputs.

   1. Argument list.

      None.

   2. COMMON blocks.

      /RPAANL/

      IRPTIM - Valid time for RPA analysis result

   3. Files.

      None.

e. Associated subroutines.

   1. Subroutine SWPCOL. Collects an RPA sweep (see 2.4.4.2.1).

   2. Subroutine RPAALG. Process the RPA sweep (see 2.4.4.3.1).

f. Interfaces.

1. Called by Subroutine PROCES.

2. Uses COMMON blocks DFRAME, RPACON, SATEPH, RPAANL, and EPANL.

2.4.4.3.1 Subroutine RPAALG.

a. Function.

This subroutine processes the raw RPA current vs. voltage curve. Successful analysis will result in the determination of ion density, ion temperature, ram direction drift velocity, and sensor potential.

b. Inputs.

1. Argument list.

C	-	Collected RPA sweep currents
GMLAT	-	Geomagnetic latitude of spacecraft
IRPDGP	-	Diagnostic output flag 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files
N	-	Number of points in sweep
SPIN	-	Sensor potential input default value
V	-	Collected RPA sweep voltages
VAPER	-	Aperture voltage (VBIAS + VIP)
VS	-	Spacecraft velocity

2. COMMON blocks.

/CONSTS/

MP	-	Mass of a proton

/RPACON/

DVLIM	-	Solution test criteria, voltage step difference
IRPMDV	-	Maximum number of derivatives to use in fit
IRPSDV	-	Starting number of derivatives to use in fit
IRPWID	-	Characteristic width for relative minima
RPXVB	-	Helium test extra bias voltage
VARLIM	-	Solution test criteria, variance limit

3. Files.

None.

c. Processing.

1. Initialize variables as necessary.

2. Calculate linear current from the input log values and validate the sweep.

3. Calculate the fractional currents, e.g., the voltages at which the current is 0.9*max, 0.8*max, ..., 0.1*max.

4. Determine the usable voltage range of the sweep.

5. Locate the minima in the first derivative of the current-voltage curve (the sweep).

6. Determine the ion mode (one ion or two ions) and select either hydrogen or helium as the predominant light ion. Oxygen is assumed to be the heavy ion or the single ion case species.

7. Calculate the plasma parameters based upon the selected ion mode, the ion masses, and the current and voltage at the determined sweep derivative minima.

8. Calculate a theoretical sweep based upon the calculated plasma parameters and the known instrument configuration.

9. Compare the measured sweep against the theoretical one. If the agreement is good, keep this solution. If the solution is not within the user-specified test criteria, vary the location of the heavy ion minima slightly and return to step 7. If the solution meets the specified criteria, store the analysis results and exit.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

   /RPAANL/

IRPAR	-	RPA processor analysis results flag
IRPLIF	-	RPA analysis light ion flag
NRPASG	-	Number of good RPA sweeps processed this readout
NRPAST	-	Number of RPA sweeps attempted this readout
RPAUR	-	RPA ram direction drift velocity
RPEDEN	-	RPA estimated electron density
RPHDEN	-	RPA light ion density
RPITMP	-	RPA ion temperature (heavy ion)
RPODEN	-	RPA heavy ion density
RPVPOT	-	RPA sensor potential

3. Files.

   None.

e. Associated subroutines.

1. Subroutine CHKSWP. Cleans up and verifies the RPA sweep (see 2.4.4.3.2).

2. Subroutine XPOS. Locates the voltages at which specified intervals in the sweep current are to be found (see 2.4.4.3.5).

3. Subroutine SSLOPE. Locate local minima in the first derivative of the current vs. voltage curve (see 2.4.4.3.6).

4. Subroutine PLASMA. Calculate the environmental parameters from the engineering type data provided by the instruments (see 2.4.4.3.7).

5. Subroutine RPASWP. Calculate the expected instrument response to a specified plasma environment (see 2.4.4.3.8).

6. Subroutine COMPAR. Calculate the variance between the measured sweep and the theoretical sweep specified by RPASWP (see 2.4.4.3.9).

7. Subroutine ANLSAV. Store a potential solution in a specified array (see 2.4.4.3.10).

8. Subroutine FITLIN. Perform a least squares linear fit to a set of points (see 2.4.8.6).

f. Interfaces.

1. Called by Subroutine RPAPRC.

2. Uses COMMON blocks CONSTS, RPACON, and RPAANL.

2.4.4.3.2 Subroutine CHKSWP.

a. Function.

Clean up and verify the raw sweep data. This routine attempts to eliminate single point anomalies, determine if the sweep meets the established range criteria, and eliminate any useless data at the end of the sweep. This subroutine is mostly based upon the AFGL routine CHECKDA.

b. Inputs.

1. Argument list.

C	-	Sweep currents (linear)
CL	-	Sweep currents (log)
IRPDGP	-	RPA diagnostic output flag 0: No diagnostic output 1: Generate binary diagnostic file 2: Generate ASCII diagnostic file 3: Generate both binary and ASCII diagnostic files
NUM	-	Number of points in sweep
V	-	Sweep voltages

2. COMMON blocks.

/RPACON/

IRPMIN	-	Minimum number of data points needed for analysis
IRPWID	-	Characteristic width for relative minima
RPCRNG	-	Required current range for an analysis
RPMAXV	-	RPA sweep maximum voltage
RPMINV	-	RPA sweep minimum voltage
RPMPCT	-	Maximum current lower limit (% of sensitivity)
RPNSD1	-	Acceptable first noise level decrease in current
RPNSD2	-	Acceptable second noise level decrease in current
RPNSUP	-	Acceptable noise level increase in current
RPSENS	-	RPA instrument sensitivity

3. Files.

None.

c. Processing.

1. Locate a valid sweep starting point. The start must be at least a specified percent level above the instrument sensitivity. Successive points must be within expected noise limits of each other. In performing the specified checks in this routine, the sweep is scanned from low voltage to high voltage (maximum current to minimum current).

2. Check for monotonically increasing voltages and other indications of trash such as currents less than the instrument sensitivity and sweep voltages outside the specified instrument sweep range.

3. Ideally, clean sweeps should be monotonically decreasing in current as well. However, since the environment is not stable during the entire sweep period and there is round-off error in encoding the data into the telemetry stream, we must allow for current variations in both directions within the user-specified limits of geophysical and telemetry noise.

4. Locate the maximum and minimum currents. The maximum current is the mean value of a specified number of points at the low voltage end of the sweep.

5. Test the current range and remaining number of points to determine whether the sweep is usable.

d. Outputs.

1. Argument list.

C	-	Sweep currents (linear)
CL	-	Sweep currents (log)
CMAX	-	Calculated maximum current level
GOOD	-	Sweep quality flag
IMAX	-	Array location of max current
NUM	-	Number of points remaining in sweep
V	-	Sweep currents
VCMAX	-	Voltage at sweep maximum current location

2. COMMON blocks.

   None.

3. Files.

   None.

e. Associated subroutines.

1. Subroutine RPDIAG. Provide diagnostic messages to the user (see 2.4.4.3.3).

2. Subroutine MEANSD. Calculate the mean and standard deviation of a portion of a table of values (see 2.4.4.3.4).

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses COMMON block RPACON.

2.4.4.3.3 Subroutine RPDIAG.

a. Function.

Provide diagnostic messages to the user concerning problems encountered during RPA data processing. Additionally, provide a copy of the raw RPA sweep for further diagnostic study at a later time.

b. Inputs.

1. Argument list.

VOLT	-	RPA raw sweep voltage array
CURR	-	RPA raw sweep current array
MSG	-	Diagnostic message number
IDIAG	-	RPA diagnostic output flag
		0: No diagnostic output
		1: Generate binary diagnostic file
		2: Generate ASCII diagnostic file
		3: Generate both types of diagnostic files

2. COMMON blocks.

/DFRAME/

IDTIME	-	Data frame time (seconds since midnight GMT)
IQFRME	-	Data frame quality flag
VBIAS	-	Instrument bias voltage

/LUDIAG/

LURPAA	-	Logical file number for RPA ASCII diagnostic file
LURPAB	-	Logical file number for RPA binary file

/PROCON/

IFLT	-	Vehicle flight number

/RPAANL/

IRPLIF	-	Light ion type flag
IRPTIM	-	RPA sweep time (seconds since midnight GMT)
RPAUR	-	RPA ram direction drift velocity
RPITMP	-	RPA ion temperature
RPHDEN	-	RPA light ion density
RPODEN	-	RPA heavy ion density
RPVPOT	-	RPA sensor potential

3. Files.

None.

c. Processing.

1. The design is for a short, single line, diagnostic concerning any problem encountered in the sweep analysis so that any trends or repetitive data problems can be isolated.

2. On first call, open the requested diagnostic files. The binary file will contain the raw RPA sweep and any analysis accomplished by this processor. The ASCII file will contain any diagnostic messages relating to the processor analysis.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

None

3. Files.

Diagnostic print to IESRPAPRT and binary output to file IESRPADIAG.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from RPAALG and CHKSWP.

2. Uses COMMON blocks DFRAME, LUDIAG, PROCON, RPAANL, and RPACON.

2.4.4.3.4 Subroutine MEANSD.

a. Function.

Calculate the mean and standard deviation of the first specified number of elements in a table of values.

b. Inputs.

1. Argument list.

M	-	Number of values to use, starting with the first value in the table
Y	-	Value table

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Calculate the mean value of the specified points.

2. Calculate the standard deviation.

d. Outputs.

1. Argument list.

SIG	-	Standard deviation
YM	-	Mean value

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CHKSWP.

2. Uses no COMMON blocks.

2.4.4.3.5 Subroutine XPOS.

a. Function.

Locate the voltages at which specified intervals in the sweep current are to be found.

b. Inputs.

1. Argument list.

K	-	Number of current intervals
N	-	Number of table values
X	-	Ordinate table
Y	-	Function table
YMAX	-	Maximum function value

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Calculate the required current level.

2. Scan the sweep until the sweep data points on either side of the current of interest are located. Remember that the sweep is assumed to be monotonically decreasing currents.

3. Perform a linear interpolation to specify the voltage at which the current level occurs.

4. Make sure we found a value for each current value (i.e., test the current range). A bad sweep flag is returned if values cannot be found. If it is specified that the current is to be divided into tenths, the sweep current must drop to one tenth the max current or a bad sweep flag will be returned.

d. Outputs.

1. Argument list.

DFX	-	Differences between ordinate fractional values
FX	-	Ordinates for fractional values
FY	-	Fractional function values
GOOD	-	Result flag
JF	-	Data index location of fractional values

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses no COMMON blocks.

2.4.4.3.6 Subroutine SSLOPE

a. Function.

Locate local relative minima in the first derivative of the current vs. voltage sweep.

b. Inputs.

1. Argument list.

C	-	Sweep current values
LCMAX	-	Array location of maximum current
NUM	-	Number of current values
NUMFIT	-	Number of data points to use in linear fit
V	-	Sweep voltage values

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Calculate the first derivative at each point in the sweep based upon the specified number of points to use in the linear fit. Derivative calculation in all steps is based upon a least squares linear fit to successive data points.

2. Locate the local minima by examination of the second derivative using the linear fit routines in the first derivative values.

3. Store the location of each local minima found.

d. Outputs.

1. Argument list.

BI	-	Intercept for linear fit at the local slope minima
KS	-	Data point location of the local slope minima
NS	-	Number of local slope minima found
S	-	Slope of current curve at the local minima

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses no COMMON blocks.

2.4.4.3.7 Subroutine PLASMA.

a. Function.

Calculate the environmental parameters from the engineering type data provided by the Retarding Potential Analyzer.

b. Inputs.

1. Argument list.

CION	-	Currents at constituent ion local slope minima
CMAX	-	Maximum sweep current

GMLAT	-	Geomagnetic latitude
MASS	-	Constituent ion atomic mass units
MODE	-	Ion processing mode
SLOPE	-	Slope at heavy ion local slope minima
SPIN	-	Default sensor potential for one ion mode
VION	-	Voltages at ion constituent slope minima
VS	-	Spacecraft velocity

2. COMMON blocks.

/CONSTS/

E	-	Electron charge
K	-	Boltzmann's constant
MP	-	Mass of a proton

/PIDEG/

PI	-	π

/RPACON/

RPLMAX	-	Magnetic latitude limit for drift velocity = 0 assumption
RPAREA	-	RPA instrument aperture area
RPTRAN	-	RPA instrument transparency

3. Files.

None.

c. Processing.

1. Test the slope value determined at the local minima of the first derivative of the current vs. voltage curve specified by the RPA sweep. If a valid value has been specified, proceed with the calculations and test for negative current values.

2. If the ion mode is two, calculate the parameters for both the heavy and the light ion species.

3. If the ion mode is one, only the heavy ion is present. If only one local minimum in the derivative of the sweep current was found, we can not calculate all parameters of interest unambiguously. Therefore, we must make some assumptions.

   (a) If we are in geomagnetic auroral or polar latitudes, use an estimated sensor potential and calculate the ram direction drift velocity.

   (b) If we are in geomagnetic mid-latitudes, assume that the ram direction drift velocity is zero and calculate the sensor potential.

d. Outputs.

1. Argument list.

DH	-	Heavy ion density
DL	-	Light ion density
DRVEL	-	Average ram direction drift velocity
RPAPOT	-	RPA sensor potential
TI	-	Heavy ion temperature

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses COMMON blocks CONSTS, PIDEG, and RPACON.

2.4.4.3.8 Subroutine RPASWP

a. Function.

Calculate the expected instrument response to a specified plasma environment based upon the engineering parameters of the RPA instrument.

b. Inputs.

1. Argument list.

MI	-	Atomic masses of constituent ions
NI	-	Number densities of ion constituents
NVOLTS	-	Number of voltages at which currents desired
SCPOT	-	Sensor potential
TI	-	Ion temperature
VNORM	-	Normal component of velocity between sensor and ions
VOLTS	-	Voltages at which currents desired

2. COMMON blocks.

/RPACON/

RPAREA	-	RPA instrument aperture area
RPTRAN	-	RPA instrument transparency

/CONSTS/

E	-	Electron charge
K	-	Boltzmann's constant

/PIDEG/

PI	-	π

3. Files.

None.

c. Processing.

1. Calculate the instrument parameters.

2. Calculate most probable speed of the ion constituents based upon the provided environment parameters and the ratio of that speed to the spacecraft velocity.

3. Calculate the current contribution of each constituent ion based upon the previously calculated parameters, the sensor potential relative to the ambient plasma, and the specified ion densities.

d. Outputs.

1. Argument list.

CUR	-	Theoretical currents at requested voltages

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from RPAALG.

2. Uses COMMON blocks CONSTS, PIDEG, and RPACON.

2.4.4.3.9 Subroutine COMPAR.

a. Function.

Calculate the variance between the measured sweep and the theoretical sweep specified by Subroutine RPASWP for both the linear and the log current values.

b. Inputs.

1. Argument list.

C	-	Measured sweep currents (linear)
CL	-	Measured sweep currents (log)
CMIN	-	Minimum sweep current
CT	-	Theoretical sweep currents
LH	-	Location of heavy ion slope minimum
MODE	-	Ion processing mode (1=single ion, 2=two ions)
N	-	Number of data points in sweep

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. If the current values appear valid:

(a) Calculate the differences between the actual and theoretical linear and log current values.

(b) Sum up the square of the differences.

2. Divide by the number of values summed to calculate the variance.

d. Outputs.

1. Argument list.

DCI	-	Current differences near heavy ion sweep slope minimum
SUM	-	Sum of current differences (linear)
SUML	-	Sum of current differences (log)

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses no COMMON blocks.

2.4.4.3.10 Subroutine ANLSAV.

a. Function.

Store a potential RPA sweep solution in a specified array.

b. Inputs.

1. Argument list.

C2	-	Current at heavy ion slope minima
DH	-	Heavy ion density
DL	-	Light ion density
DV	-	Ram direction drift velocity
M1	-	Light ion AMU
MODE	-	Ion processing mode
S	-	Slope at heavy ion slope minima
SP	-	Sensor potential
SUM	-	Linear variance
SUML	-	Log variance
TI	-	Ion temperature
V2	-	Voltage at heavy ion slope minima

2. COMMON blocks.

None.

3. Files.

None.

c. Processing.

1. Increment the array index counter.

2. Store the individual values.

d. Outputs.

1. Argument list.

AA	-	Analysis results array
NSOL	-	Number of potential solutions in the analysis results array

2. COMMON blocks.

None.

3. Files.

None.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine RPAALG.

2. Uses no COMMON blocks.

2.4.4.4 Subroutine DMPRC.

a. Function.

Convert voltages from the Drift Meter (DM) into ion densities and horizontal and vertical cross track ion drift velocities. Figure 11 is a hierarchy chart for this module. The DM data processing algorithm is described in Appendix B.3.

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag	
NEWREV	-	New readout-REV flag	
GRESET	-	General reset flag	
THEEND	-	End-of-data flag	
IDODM	-	DM process control switch	
		0: OFF, 1: ON	
		bit 1 (LSB):	Normal mode
		bit 2:	H+ mode
		bit 3:	DM FIBA
IDIAG	-	Diagnostic output flag	

DMPRC:	process driftmeter (DM) data
. DMINFO:	determine current state of DM sensor
.. DMCMD:	interpret DM command value
... DMSWCH:	set processing status for DM mode change
... PRNTON:	set PRINTAWAY status for conclusion of processing
.. PRNTON:	set PRINTAWAY status for conclusion of processing
. PRNTON:	set PRINTAWAY status for conclusion of processing
. DMVEL:	convert DM voltages to ion drift velocity and angle of arrival
. PRMRNG:	check environment parameters against allowed ranges
. DMDEN:	calculate ion density for DM log level data
. DMFIBA:	convert DM filter band values to power spectral density values
.. OPNDOF:	initialize diagnostic output file
... BITON:	test for bit setting
... FILERR:	determine and report file error status
... PRNTON:	set PRINTAWAY status for conclusion of processing
. HPMODE:	report DM H+ mode data
.. OPNDOF:	initialize diagnostic output file
... BITON:	test for bit setting
... FILERR:	determine and report file error status
... PRNTON:	set PRINTAWAY status for conclusion of processing
. DMDIAG:	report DM processing diagnostics
.. OPNDOF:	initialize diagnostic output file
... BITON:	test for bit setting
... FILERR:	determine and report file error status
... PRNTON:	set PRINTAWAY status for conclusion of processing
.. FILERR:	determine and report file error status

Figure 11 DMPRC module hierarchy chart

2. COMMON blocks.

/CONSTS/

E	-	Electron charge (coulombs)

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
ICYCLE	-	Data frame cycle ID (1 or 2)
IQFRME	-	Data frame quality flag
DMOFF	-	12 offset voltages from the DM sensor
MONDSM	-	DSM command from last cycle 2 data frame
DMLL	-	Log-level voltage from DM sensor cycle 1: LLA cycle 2: LLB
DMFILT	-	DM filter values

/DMCON/

CMK1	-	Velocity calculation constant
DMAEFF	-	Effective area of DM sensor (cm^2)

DMK2	-	Density calculation constant
DMOOFF	-	Nominal offset data zero voltage (volts)
DMAG	-	Gain of LLA amplifier
DMBG	-	Gain of LLB amplifier
DMBOFF	-	Offset of LLB amplifier (volts)
ANGMAX	-	Maximum allowed angle-of-arrival (deg)
DMSH	-	Sign of horizontal velocity (± 1)
DMSV	-	Sign of vertical velocity (± 1)
IDMH	-	Location marker for horizontal velocity (0 or 1)
IDMV	-	Location marker for vertical velocity (1 or 0)
NOLL	-	LL reliability flag
NOHP	-	H+ mode reliability flag
IDMUR	-	RPA Ram velocity use flag

/RPAANL/

RPAUR	-	Ram ion drift velocity (m/s)

/SATEPH/

VSC	-	Satellite orbital velocity (m/s)

/SCSOH/

DMV0	-	DM sensor offset voltage

c. Processing.

1. If no more data, return to Subroutine PROCES.

2. Reset processing flags for a new satellite, new REV, or a general reset and decode DM processing switches.

3. Invoke Subroutine DMINFO to determine the status of the DM sensor and break out the cycle dependent DM data in the current data frame.

4. If the data frame is bad, or if DM processing is in a wait status, return to Subroutine PROCES.

5. Use the nominal offset zero voltage (DMOOFF) for the offset measurements.

6. Calculate the total velocity of the ions with respect to the satellite using the RPA analysis ram velocity when applicable.

7. If normal mode data are to be processed:

   (a) Convert each pair of horizontal/vertical offset voltages into horizontal and vertical drift velocities (Function DMVEL). Calculate the average horizontal and vertical drift velocities.

   (b) Save the tangent of the angle-of-arrival (TANA) from the offset voltage nearest (in the data frame) to the DMLL data for use in calculating ion density from the DMLL data.

   (c) If the DM repeller voltage is near zero, invoke Function DMDEN to calculate the ion density

from the DMLL voltage.

(d) If in H+ mode, calculate one set of drift velocities from the first pair of horizontal/vertical offset voltages. Set the average velocities to these values.

8. If in FIBA mode, and FIBA mode data are to be processed, invoke Subroutine DMFIBA to process the data.

9. If in H+ mode, and H+ mode data are to be processed, invoke Subroutine HPMODE to process the data.

10. If diagnostic output was requested, invoke Subroutine DMDIAG to produce the output.

11. Return to Subroutine PROCES.

d. Outputs.

1. COMMON blocks.

/DMANL/

DMUH	-	Horizontal ion drift velocities (6) (m/s)
DMAUH	-	Average horizontal ion drift velocity (m/s)
DMUV	-	Vertical ion drift velocities (6) (m/s)
DMAUV	-	Average vertical ion drift velocity (m/s)
DMIDEN	-	Ion density (ion/cm^3)
IDMMDE	-	DM sensor mode 0=Normal; 1-8=H+; 9=FIBA
IDMEF	-	DM analysis error flag
IDMAR	-	DM analysis results flag

e. Associated subroutines.

1. Function BITON. Determine if a specific bit in a word is set (see 2.4.8.2).

2. Function DMINFO. Determine the status of the DM sensor and extract cycle-dependent data from the data frame (see 2.4.4.4.1).

3. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console message error flag (see 2.4.8.11).

4. Function DMVEL. Calculate ion drift velocity (see 2.4.4.4.4).

5. Function DMDEN. Calculate ion density (see 2.4.4.4.5).

6. Subroutine DMFIBA. Process filter data (see 2.4.4.4.6).

7. Subroutine HPMODE. Process H+ mode data (see 2.4.4.4.7).

8. Subroutine DMDIAG. Generate diagnostic output (see 2.4.4.4.8).

9. Function PRMRNG. Determine if a parameter is within the allowed range (see 2.4.8.10).

f. Interfaces.

1. Called from Subroutine PROCES.

2. Uses COMMON blocks CONSTS, DFRAME, DMANL, DMCON, PIDEG, RPAANL, SATEPH, and SCSOH.

2.4.4.4.1 Subroutine DMINFO.

a. Function.

Determine the present status of the DM sensor and extract cycle dependent data from the data frame.

b. Inputs.

1. Argument list.

RESET	-	Reset flag (LOGICAL)
ICYCLE	-	Data frame cycle ID (1 or 2)
ICCNTR	-	Data frame cycle counter (1 - 1024)
IQFRME	-	Data frame quality flag
IDTIME	-	Data frame time
MONDSM	-	DSM command from last cycle 2 data frame
DMLL	-	Log-level voltage from DM sensor cycle 1: LLA cycle 2: LLB
IDMMDE	-	DM sensor mode 0=Normal; 1-8=H+; 9=FIBA
IHPCYC	-	H+ mode cycle counter
NOLL	-	LL reliability flag
DMRV	-	DM repeller voltage

c. Processing.

1. If a reset is called for (new satellite, readout REV, or initialization), set wait flag and initialize wait and H+ mode counters to prevent processing of data before DM operation mode can be determined.

2. Decrement mode switch cycle counter. When counter = 0, switch to new mode, release wait, and initialize H+ cycle counter.

3. If in H+ mode, increment H+ cycle counter. (Range = 1 - 4)

4. If the data frame is bad, return to Subroutine DMPRC.

5. If this is an even data cycle (2, 4, ..), invoke Subroutine DMCMD to process DM command information

6. If the LL data is in normal range, the reliability flag is OK (NOLL=0), and we are not in a wait state, save the LL data for processing.

7. If a reset call, place the processing into wait status.

8. When in a wait state, turn on the print (Subroutine PRNTON), alert the user when the wait begins, when it has been waiting for 20 seconds, and every 10 minutes thereafter.

9. Return to Subroutine DMPRC.

d. Outputs.

1. Argument list.

IDMMDE	-	DM sensor mode 0=Normal; 1-8=H+; 9=FIBA
IHPCYC	-	H+ mode cycle counter
DMRV	-	DM repeller voltage
DMLLA	-	LLA voltage (cycle 1)
LLAOK	-	LLA quality flag (cycle 1)
DMLLB	-	LLB voltage (cycle 2)
LLBOK	-	LLB quality flag (cycle 2)
WAIT	-	Wait status flag

e. Associated subroutines.

1. Subroutine DMCMD. Process the DM command (see 2.4.4.4.2).

2. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

f. Interfaces.

1. Called from Subroutine DMPRC.

2. Uses no COMMON blocks.

2.4.4.4.2 Subroutine DMCMD.

a. Function.

Process DM sensor commands.

NOTE: DMCMD processing is specific to **IES2 and IES2A** only. When IES3 comes on line, DMCMD may need to be modified to handle any IES3 specific DSM command codes. If it becomes necessary to process IES data, DMCMD must be modified to handle the IES specific DM command codes.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag
MONDSM	-	DSM command (from data frame)

DMRV	-	DM repeller voltage (volts)
IDMMDE	-	DM sensor mode (0 - 9)
		0=Normal; 1-8=H+; 9=FIBA
ICCNTR	-	Data cycle counter (1 - 1024)

2. COMMON blocks.

/RGSTAT/

IDMSTA	-	DM mode status
IRPSTA	-	Repeller level status
ISTSTA	-	Subcom processing status

c. Processing.

1. If a reset is required, set last command to -1.

2. If the current DM command matches previous one, set the new command flag (NEWCMD) to .FALSE. and jump to step 4.

3. If there is a new command and it is a valid DM command:

(a) set new command flag to .TRUE. and set last command to command value.

(b) If the command is a 'normal' or an H+ mode command, invoke Subroutine DMSWCH to determine which mode has been selected and at which cycle it will begin.

(c) If the command is a DM FIBA mode command, invoke Subroutine DMSWCH to determine the proper mode and to determine at which cycle it will begin.

(d) If the command is to change the repeller voltage, and:

(1) we are currently in H+ mode, invoke Subroutine DMSWCH to determine the proper cycle for the repeller voltage and mode switch.

(2) we are in normal or FIBA mode, calculate the new repeller voltage, alert the user to the change, and set the wait flag to skip processing for this cycle and resume processing on the next cycle with the new repeller voltage.

(e) If the command is to change the 'wiggle' level (H+ mode) setting and:

(1) we are in FIBA mode, this is an abnormal command for FIBA mode. Alert the user and disregard the command.

(2) calculate the H+ mode from the command value, invoke Subroutine DMSWCH to determine the proper cycle for the mode change to take place, and alert the user of the mode change.

(f) Otherwise, this was a 'spares' command. Alert the user and disregard the command.

4. If the current 16 second subcom status is available:

   (a) if the subcom status does not match current command mode status, alert the user, set the DM mode status to that of the subcom, and set the wait flag to skip this cycle and resume processing in the new mode on the next cycle.

   (b) set the repeller voltage from the subcom information.

5. If the return status is not zero, invoke Subroutine PRNTON to turn on the print.

6. Return to Subroutine DMINFO.

d. Outputs.

1. Argument list.

DMRV	-	DM repeller voltage
IDMMDE	-	DM sensor mode 0=Normal; 1-8=H+; 9=FIBA
IHPCYC	-	H+ cycle counter
NCTOGO	-	Number of cycles to wait for mode change.
NXTMDE	-	Mode which takes effect when NCTOGO = 0
WAIT	-	Processing wait flag
ISTAT	-	Return status

e. Associated subroutines.

1. Subroutine DMSWCH. Determine when a DM mode switch will be activated (see 2.4.4.4.3)

2. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

f. Interfaces.

1. Called from Subroutine DMINFO.

2. Uses COMMON block RGSTAT.

2.4.4.4.3 Subroutine DMSWCH.

a. Function.

Set the proper parameters for a Driftmeter mode switch.

b. Inputs.

1. Argument list.

ICCNTR	-	Data cycle counter (1 - 1024)
IDMMDE	-	Drift meter sensor mode (0 - 9) 0=Normal; 1-8=H+; 9=FIBA

NXTMDE	-	Next commanded DM mode

c. Processing.

1. Calculate the number of data cycles until the next 16 secsync pulse. Note that H+ and FIBA modes activate and deactivate only on 16 second (secsync) cycle boundaries.

2. Determine whether to continue processing in the current mode, or to suspend processing until the mode switch occurs.

d. Outputs.

1. Argument list.

NCTOGO	-	Number of data cycles until mode switch occurs
WAIT	-	Suspend DM processing flag (LOGICAL): .TRUE.: suspend processing until mode switch .FALSE.: continue in current mode until switch

e. Associated subroutines

None.

f. Interfaces.

1. Called from Subroutine DMCMD.

2. Uses no COMMON blocks.

2.4.4.4.4 Function DMVEL.

a. Function.

Calculate the ion drift velocity.

b. Inputs.

1. Argument list.

R	-	Offset current ratio from DM sensor (volts)
DMK1	-	Drift velocity calculation constant
VOFF	-	Zero voltage for offset measurement (volts)
VTOT	-	Total ion velocity w.r.t. satellite (m/s)
TANMAX	-	Tangent of maximum allowed angle-of-arrival

c. Processing.

1. Calculate the tangent of the ion angle-of-arrival.

2. If the angle is within the allowed range, calculate the drift velocity. If not, set the drift velocity and the tangent to the missing data flag (the TANA variable is also set to XMSSNG as it is used

in the ion density calculation).

3. Return to Subroutine DMPRC.

d. Outputs.

1. Function value.

DMVEL	-	Ion drift velocity (m/s)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine DMPRC.

2. Uses no COMMON blocks.

2.4.4.4.5 Function DMDEN.

a. Function.

Convert the voltage from one of the DM log-level amplifiers into an ion density.

b. Inputs.

1. Argument list.

DMLL	-	LLA/LLB voltage
LLOK	-	Log-level quality flag (set in DMINFO)
TANA	-	Tangent of the latest angle-of-arrival
DMG	-	Amplifier gain
DMO	-	Amplifier offset
CONST1	-	E $\times$ DMAEFF $\times$ 10^6
DMK2	-	Density calculation constant
VTOT	-	Total velocity of ions w.r.t. satellite (m/s)

c. Processing.

1. If the LL data and the angle-of-arrival data are good, calculate the ion density. Otherwise, set the density to the missing data flag.

2. Return to Subroutine DMPRC.

d. Outputs.

1. Function value.

DMDEN	-	Ion density (ion/cm^3)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine DMPRC.

2. Uses no COMMON blocks.

2.4.4.4.6 Subroutine DMFIBA.

a. Function.

Convert the voltages from the Driftmeter filter comb (FIBA) into power spectral density (PSD) estimates at the specified (DMSFLT) frequencies.

Note: The DM filters were added to the telemetry for IES2 and subsequent configurations. No specific processing algorithms or techniques have been defined for this data. The processing performed here is based upon the SM filter processing. The results are written to an ASCII file for post processing analysis, and not used elsewhere within the APGA program.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag
IDTIME	-	Data frame time
ICYCLE	-	Data frame cycle indicator (1 or 2)
DMFILT	-	Array of DM filter voltages (cycle 1 only)
DMOFF	-	DM offset voltage data
WB2RNG	-	DM wideband 2 range
VTOT	-	Total ion ram velocity (m/s)

2. COMMON blocks.

/DMANL/

IDMMDE	-	DM sensor mode indicator (0 - 9) 0=Normal; 1-8=H+; 9=FIBA

/DMCON/

DMFBW	-	DM filter comb bandwidths (Hz)
DMFG	-	DM filter amplifier gain
DMFOFF	-	DM filter comb offset voltages
DMVTAB	-	DM range data table (volts)
NDMFLT	-	Number of DM filter values

/LUDIAG/

LUFIBA - Logical unit for DM filter print file

c. Processing.

1. On the first call, invoke Subroutine OPNDOF to open the output file. If the open is not successful, alert the user.

2. If the output file is not open or the output has been turned off (ASCII = .FALSE.), return to Subroutine DMPRC.

3. If a reset has been ordered, write the header record to the output file.

4. Write the time, data cycle and DM offset voltages to the output file.

5. If this is an odd data cycle:

   (a) attempt to find the WB2 range sensitivity bin.

   (b) if no bin was found, there is a WB2 range error. Alert the user.

   (c) if the WB2 range is okay, calculate the PSD estimates from the DM filter data for which the bandwidth is non-zero as follows:

      (1) calculate the RMS current variation offset.

      (2) calculate the RMS current variation.

      (3) calculate the RMS density variation.

      (4) calculate the PSD estimates.

   (d) if there is a WB2 range error, load the missing value ID (-1.0×10^{37}) into the PSD array.

6. Write the DM mode, filter voltages, and PSD estimates to the output file.

7. Return to Subroutine DMPRC.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

   /DMANL/

   DMFPSD - Array of PSD estimates.

3. Files.

IESDMFIBA

e. Associated subroutines.

1. Subroutine OPNDOF. Open diagnostic files (see 2.4.4.8)

f. Interfaces

1. Called from Subroutine DMPRC.

2. Uses COMMON blocks DMANL, DMCON, and LUDIAG.

2.4.4.4.7 Subroutine HPMODE.

a. Function.

Ultimately, the function of this subroutine will be to process H+ mode data from the DM sensor. No specific processing algorithms or techniques have been defined for this data. This subroutine merely reformats the input voltages and writes them to an ASCII diagnostic output file (IESDMHPMODE).

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag
IDTIME	-	Data frame time (seconds since midnight)
IDMMDE	-	DM sensor mode indicator (0 - 9) 0=Normal; 1-8=H+; 9=FIBA
IHPCYC	-	H+ mode cycle counter (1 - 4)
DMOFF	-	DM sensor offset voltages

2. COMMON blocks.

/LUDIAG/

LUDMHP	-	Logical unit for H+ mode diagnostic output

c. Processing.

1. On first call, open the output file (Subroutine OPNDOF). If a problem was encountered in opening the file, alert the user.

2. If this output has been turned off or the open was not successful (ASCII=.FALSE.), return to Subroutine DMPRC.

3. If a reset, write out the header.

4. Reformat the DMOFF voltages to integer and write out to the file.

5. Return to Subroutine DMPRC.

d. Outputs.

1. Files.

IESDMHPMODE. Figure 12 shows a sample of the output.

a: Time of record (seconds since midnight)
b: DM operating mode
1: Low wiggle, 0.0 volts starting potential
2: High wiggle, 0.0 volts starting potential
3: Low wiggle, 1.0 volts starting potential
4: High wiggle, 1.0 volts starting potential
5: Low wiggle, 2.0 volts starting potential
6: High wiggle, 2.0 volts starting potential
7: Low wiggle, 3.0 volts starting potential
8: High wiggle, 3.0 volts starting potential
c: DM H+ mode cycle counter (1-4)
d-o: H+ mode data samples for one-data-second (raw data)

Figure 12 Format and contents of the DM H+ mode output file

e. Associated subroutines.

1. Subroutine OPNDOF. Open diagnostic output files (see 2.4.8.9).

f. Interfaces.

1. Called from Subroutine DMPRC.

2. Uses COMMON block LUDIAG.

2.4.4.4.8 Subroutine DMDIAG.

a. Function.

Construct diagnostic output files for the DM data processing module.

b. Inputs.

1. Argument list.

IDTIME	-	Time of current data frame (seconds since midnight)
MONDSM	-	DSM command from last cycle 2 data frame
DMOFF	-	12 offset voltages from the DM sensor
DMLL	-	Log-level voltage from DM sensor: cycle 1: LLA cycle 2: LLB
DMEQP	-	DM equipment information: cycle 1: DM offset voltage monitor (volts) cycle 2: DSM electronics temperature
VTOT	-	Total ion velocity w.r.t. satellite (m/s)
IDIAG	-	Diagnostic print flag: 0: No diagnostic output 1: Binary diagnostic output only 2: ASCII diagnostic output only 3: Both binary and ASCII output

2. COMMON blocks.

/DMANL/

DMUH	-	Horizontal ion drift velocities (6) (m/s)
DMAUH	-	Average horizontal ion drift velocity (m/s)
DMUV	-	Vertical ion drift velocities (6) (m/s)
DMAUV	-	Average vertical ion drift velocity (m/s)
DMIDEN	-	Ion density (ion/cm^3)
IDMMDE	-	DM sensor mode (0 - 9) 0=Normal; 1-8=H+; 9=FIBA
IDMEF	-	DM analysis error flag
IDMAR	-	DM analysis results flag

/DMCON/

DMK1	-	Velocity calculation constant
DMAEFF	-	Effective area of DM sensor (cm^2)
DMK2	-	Density calculation constant
DMOOFF	-	Nominal offset data zero voltage (volts)
DMAG	-	Gain of LLA amplifier
DMAOFF	-	Offset of LLA amplifier (volts)
DMBG	-	Gain of LLB amplifier
DMBOFF	-	Offset of LLB amplifier (volts)
ANGMAX	-	Maximum allowed angle-of-arrival (deg)
DMSH	-	Sign of horizontal velocity ($\pm$1)

DMSV	-	Sign of vertical velocity (± 1)
IDMH	-	Location marker for horizontal velocity (0 or 1)
IDMV	-	Location marker for vertical velocity (0 or 1)
NOLL	-	LL reliability flag
NOHP	-	H+ mode reliability flag
IDMUR	-	RPA ram velocity use flag

/LUDIAG/

LUDMA	-	Logical unit for ASCII output
LUDMB	-	Logical unit for binary output

/PROCON/

IFLT	-	Satellite flight ID

/SCSOH/

DMV0	-	DM sensor offset voltage

c. Processing.

1. On first call to this routine, open the necessary ASCII and/or binary output files (Subroutine OPNDOF).

2. If neither file could be opened, set the diagnostic print flag to zero (disabling the diagnostic output) and return to Subroutine DMPRC.

3. If an ASCII output file is requested, then

   (a) If a RESET=.TRUE. call, output the sensor processing constants and processing control information for the DM module.

   (b) If the data frame was good, write out all data input and produced in the current invocation of the DM processor. (Note: If the frame was bad, write out a note to that effect).

4. If a binary output file is requested, then if the frame was good, write the diagnostic output buffer to the output file.

5. Return to Subroutine DMPRC.

d. Outputs.

IESDMPRT. ASCII diagnostic output file (see Section 4.4.5).

IESDMDIAG. Binary diagnostic output file (see Section 4.4.5).

e. Associated subroutines.

1. Subroutine OPNDOF. Open diagnostic output files (see 2.4.8.9).

2. Subroutine FILERR. File error handling routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine DMPRC.

2. Uses COMMON blocks DMANL, DMCON, LUDIAG, PROCON, and SCSOH.

2.4.4.5 Subroutine MPPRC.

a. Function.

This routine is the driver for the routines which process data from the on-board microprocessor analyses of EP and RPA sweep data. Figure 13 is a hierarchy chart for this module. The MP data processing algorithms are described in Appendix B.6.

MPPRC:	process microprocessor (MP) data
. BITON:	test for bit setting
. MPEP:	process MP data for EP
.. COPY:	initialize array or transfer values between arrays
.. UPBITS:	unpack bits from a word into an array
.. PRMRNG:	check environment parameters against allowed ranges
. MPRPA:	process MP data for RPA
.. COPY:	initialize array or transfer values between arrays
.. UPBITS:	unpack bits from a word into an array
.. PRMRNG:	check environment parameters against allowed ranges
. MPDIAG:	report MP diagnostics, and EPPRC/RPAPRC comparison
.. BITON:	test for bit setting
.. OPNDOF:	initialize diagnostic output file
... BITON:	test for bit setting
... FILERR:	determine and report file error status
... PRNTON:	set PRINTAWAY status for conclusion of processing
.. LATLON:	calculate latitude and longitude from orbit parameters

Figure 13 MPPRC module hierarchy chart

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag
THEEND	-	End-of-data flag
IDOMP	-	MP data processing flag 0: Do not process MP data 1: Process only MP EP analyses 2: Process only MP RPA analyses 3: Process both EP and RPA analyses
IDIAG	-	Diagnostic output flag (see MPDIAG)

2. COMMON blocks.

/DFRAME/

ICCNTR	-	Sweep clock program counter
IQFRME	-	Data frame quality flag
MODEP	-	EP sensor mode

c. Processing.

1. If a new satellite or REV, set a local reset flag.

2. If the EP sensor has changed modes, write out a diagnostic message, save the new mode value, and set the local reset flag.

3. If the local reset flag is set, then

   (a) If a bad frame, return to Subroutine PROCES.

   (b) If the current frame is the start of the first full RPA sweep since the reset flag was set, determine which sweep analyses are to be processed and continue processing. Otherwise, increment the local frame counter and return to Subroutine PROCES.

4. Set a local RPA sweep-step counter from the sweep clock program counter.

5. Process the EP (Subroutine MPEP) and RPA (Subroutine MPRPA) analyses as requested by the user.

6. If this is the end of the data, return to Subroutine PROCES.

7. If user-requested, generate sweep analysis comparison files (Subroutine MPDIAG).

8. Turn off the local reset flag and return to Subroutine PROCES.

d. Outputs.

None.

e. Associated subroutines.

1. Function BITON. Determine if a specific bit in a word is set (see 2.4.8.2).

2. Subroutine MPEP. Process EP Sweep analyses from the on-board microprocessor data (see 2.4.4.5.1).

3. Subroutine MPRPA. Process RPA Sweep analyses from the on-board microprocessor data (see 2.4.4.5.2).

4. Subroutine MPDIAG. Generate diagnostic output files containing comparisons of MP and ground analyses of EP and RPA sweeps (see 2.4.4.5.3).

f. Interfaces.

1. Called from Subroutine PROCES.

2. Uses COMMON block DFRAME.

2.4.4.5.1 Subroutine MPEP.

a. Function.

Process EP sweep analysis results from the on-board microprocessor.

b. Inputs.

1. Argument list.

RESET	-	Local reset flag (see MPPRC)
THEEND	-	End-of-data flag
MPCNT	-	RPA sweep-step counter (see MPPRC)

2. COMMON blocks.

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
ICYCLE	-	Data frame cycle ID (1 or 2)
ICCNTR	-	Sweep program counter
IQFRME	-	Data frame quality flag
IEPFLG	-	MP analysis results flags cycle 1: EP sweep
VBIAS	-	V_{bias} voltage
ELMP	-	MPEP sweep analysis results cycle 1: Electron temperature cycle 2: Electron density
VPEL	-	MP EP satellite potential results

c. Processing.

1. If a reset call, reset all local flags/counters.

2. If COMMON MPEANL contains an old analysis, reinitialize it.

3. If the data frame is bad, or if both cycle 1 and cycle 2 data for the current sweep have been collected, or if there are no more data, jump down to step 9.

4. Determine the status of the EP sensor and the location in the sweep program as follows:

   (a) If a cycle 1 data frame, extract the bits from the data frame ID word which indicate (1) if the EP sensor is in a calibration sweep, (2) if the EP sensor is in a DC mode, and (3) if the EP sensor is in mode E (short sweeps).

   (b) Use the sensor status information to set the number of data frames available to extract data

for the current sweep.

(c) If in a DC mode (C or D), determine if a sweep is currently in progress and if there was a sweep in progress previously for which an analysis may be available.

5. If the sensor is in a DC mode, and there was no previous sweep for which an analysis is available, return to Subroutine MPPRC.

6. If the time for the current EP sweep analysis has not been determined, calculate the sweep center time.

7. If the current data frame is cycle 1 and the MP EP analysis data from cycle 1 have not yet been extracted, then save the electron temperature measurement, calculate the satellite potential from the sensor potential reported from the MP EP analysis, and separate out the bits from the MP EP analysis results word.

8. If the current data frame is cycle 2 and the MP EP analysis data from cycle 2 have not yet been extracted, then save the electron density parameter.

9. If we are at the end of a read interval (as set by the length-of-sweep variable), then:

(a) If the electron density parameter was found, calculate the electron density using the electron temperature if it is available, or an electron temperature of 2000° K if it is not.

(b) Check the ranges of the analysis parameters found.

(c) Set the analysis results flag to 1 if any good analysis results were found.

(d) Reset the data-acquired flags for the next go-around.

10. Return to Subroutine MPPRC.

d. Outputs.

1. COMMON blocks.

/MPEANL/

METIME	-	EP sweep center time (seconds since midnight)
MPEFLG	-	MP EP sweep analysis error flags
EDENMP	-	MP EP electron density (el/cm^3)
ETMP	-	MP EP electron temperature (°K)
SPEPMP	-	MP EP satellite potential (volts)
IMPEAR	-	MP EP sweep analysis results flag

e. Associated subroutines.

1. Function BITON. Determine if a specific bit in a word is set (see 2.4.8.2).

2. Function PRMRNG. Determine if a calculated parameter is within acceptable range (see 2.4.8.10).

3. Subroutine UPBITS. Unpack the bits from a word and transfer them to an array (see 2.4.8.13).

f. Interfaces.

1. Called from Subroutine MPPRC.

2. Uses COMMON blocks DFRAME and MPEANL.

2.4.4.5.2 Subroutine MPRPA.

a. Function.

Process RPA sweep analysis results from the on-board microprocessor.

b. Inputs.

1. Argument list.

RESET	-	Local reset flag (see MPPRC)
THEEND	-	End-of-data flag
MPCNT	-	RPA sweep-step counter (see MPPRC)

2. COMMON blocks.

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
ICYCLE	-	Data frame cycle ID (1 or 2)
IQFRME	-	Data frame quality flag
INPFLG	-	MP analysis results flags cycle 2: RPA sweep
VAPER	-	$V_{bias} + V_{IP}$ (volts)
HIMP	-	MP RPA sweep analysis results cycle 1: Light ion temperature cycle 2: Light ion density
OIMP	-	MP RPA sweep analysis results cycle 1: O+ ion temperature cycle 2: O+ ion density
VSP	-	MP RPA ram drift velocity
VPION	-	MP RPA satellite potential results

c. Processing.

1. If a reset call, reset all local flags/counters.

2. If COMMON MPIANL contains an old analysis, reinitialize it.

3. If the data frame is bad, or if both cycle 1 and cycle 2 data for the current sweep have been collected, or if there are no more data, jump down to step 7.

4. If the time for the current RPA sweep analysis has not been determined, calculate the sweep

center time.

5. If the current data frame is cycle 1 and the MP RPA analysis data from cycle 1 have not yet been extracted, then save the O+ and light ion temperatures.

6. If the current data frame is cycle 2 and the MP RPA analysis data from cycle 2 have not yet been extracted, then save the O+ and light ion densities and the ion ram drift velocity, calculate the satellite potential from the sensor potential reported from the MP RPA analysis and the V_{bias} value saved from a previous data frame, save the current V_{bias} readings, separate out the bits from the MP RPA analysis results word, and set the light-ion flag word.

7. If we are at the end of a read interval (as set by the length-of-sweep variable), then

   (a) Check the ranges of the analysis parameters found.

   (b) Set the analysis results flag to 1 if any good analysis results were found.

   (c) Reset the data-acquired flags for the next go-around.

8. Return to Subroutine MPPRC.

d. Outputs.

1. COMMON blocks.

/MPIANL/

MITIME	-	RPA sweep center time (seconds since midnight)
MPIFLG	-	MP RPA sweep analysis error flags (10)
ODENMP	-	MP RPA O+ density (ion/cm^3)
OTMP	-	MP RPA O+ temperature (°K)
HDENMP	-	MP RPA light ion density (ion/cm^3)
HTMP	-	MP RPA light ion temperature (°K)
IMPLIF	-	MP RPA light ion flag
URMP	-	MP RPA ram ion drift velocity (m/s)
SPRPMP	-	MP RPA satellite potential (volts)
IMPIAR	-	MP RPA analysis results flag

e. Associated subroutines.

1. Function BITON. Determine if a specific bit in a word is set (see 2.4.8.2).

2. Function PRMRNG. Determine if a calculated parameter is within acceptable range (see 2.4.8.10).

3. Subroutine UPBITS. Unpack the bits from a word and transfer them to an array (see 2.4.8.13).

f. Interfaces.

1. Called from Subroutine MPPRC.

2. Uses COMMON blocks DFRAME and MPIANL.

2.4.4.5.3 Subroutine MPDIAG

a. Function.

Build ASCII format diagnostic output files containing comparisons of the results of EP and RPA sweep analysis from modules EPPRC and RPAPRC with the results obtained by the on-board microprocessor analyses of the same sweeps. The output goes to files IESMPEPPRT and IESMPRPAPRT. This output may be transferred to the standard HSP file by Subroutine SUMOUT if requested by the user (see 2.4.6).

b. Inputs.

1. Argument list.

RESET	-	Local reset flag (see MPPRC)
IDIAG	-	Diagnostic output flag: 0: Do not generate diagnostic output files 1: Generate an RPA sweep analysis file 2: Generate an EP sweep analysis file 3: Generate both files

2. COMMON blocks.

/EPANL/

IEPTIM	-	Sweep center time (seconds since midnight)
EPEDEN	-	DC mode electron density (24, el/cm^3)
EPETMP	-	Electron temperature (°K)
EPVPOT	-	Satellite potential (volts)
IEPAR	-	EP analysis results flag

/LUDIAG/

LUMPEP	-	Logical unit for file IESMPEPPRT
LUMPRP	-	Logical unit for file IESMPRPAPRT

/MPEANL/

METIME	-	EP sweep center time (seconds since midnight)
MPEFLG	-	MP EP sweep analysis error flags
EDENMP	-	MP EP electron density (el/cm^3)
ETMP	-	MP EP electron temperature (°K)
SPEPMP	-	MP EP satellite potential (volts)
IMPEAR	-	MP EP sweep analysis results flag

/MPIANL/

MITIME	-	RPA sweep center time (seconds since midnight)
MPIFLG	-	MP RPA sweep analysis error flags (10)
ODENMP	-	MP RPA O+ density (ion/cm^3)
OTMP	-	MP RPA O+ temperature (°K)

HDENMP	-	MP RPA light ion density (ion/cm^3)
HTMP	-	MP RPA light ion temperature (°K)
IMPLIF	-	MP RPA light ion flag
URMP	-	MP RPA ram ion drift velocity (m/s)
SPRPMP	-	MP RPA satellite potential (volts)
IMPIAR	-	MP RPA analysis results flag

/RPAANL/

IRPTIM	-	Center time of sweep (seconds since midnight)
RPHDEN	-	Light ion density (ion/cm^3)
RPODEN	-	O+ ion density (ion/cm^3)
RPEDEN	-	Total ion density (ion/cm^3)
RPITMP	-	Ion temperature (deg K)
IRPLIF	-	Light ion flag
RPAUR	-	Ram ion drift velocity (m/s)
RPVPOT	-	Satellite potential (volts)
IRPAR	-	RPA analysis results flag

/SATEPH/

ITCKL	-	Times for C_kL analyses (8) (seconds since midnight)

c. Processing.

1. If a reset call, determine which output files need to be built. If they have not already been opened, open them (Subroutine OPNDOF). If they are open already, write a new header line to the file(s). If either file could not be opened, write out a diagnostic warning and set the file output flag to .FALSE. to disable the generation of that file.

2. If an EP or RPA sweep analysis is available in the EPANL or RPAANL COMMON blocks, save it for comparison with the MP analysis.

3. If an MP EP sweep analysis is available and the EP comparison file is to be built, then:

   (a) Calculate the GMT time and the satellite location (Subroutine LATLON) for the output file.

   (b) Check both sections of the EPPRC EP sweep analysis save array for the EP sweep corresponding to the MP EP sweep analysis just received. If no match is found, write the MP EP analysis results to the diagnostic output file. If a match is found, calculate the comparison values (MP-to-EPPRC ratio for density and temperature, MP-minus-EPPRC difference for satellite potential) and write the MP EP analysis results and the comparison values to the diagnostic output file.

4. If an MP RPA sweep analysis is available and the RPA comparison file is to be built, then:

   (a) Calculate the GMT time and the satellite location (Subroutine LATLON) for the output file.

   (b) Check both sections of the RPAPRC RPA sweep analysis save array for the RPA sweep corresponding to the MP RPA sweep analysis just received. If no match is found, write the MP RPA analysis results to the diagnostic output file. If a match is found, calculate the comparison values (MP-to-RPAPRC ratio for densities and temperature, MP-minus-RPAPRC

difference for satellite potential and ion drift velocity) and write the MP RPA analysis results and the comparison values to the diagnostic output file.

5. Return to Subroutine MPPRC.

d. Outputs.

1. Files.

IESMPEPPRT. Figure 14a is a sample of the output written to this file. The items in the various columns are described in Table 1.

IESMPRPAPRT. Figure 14b is a sample of the output written to this file. The items in the various columns are described in Table 1.

e. Associated subroutines.

1. Function BITON. Determine if a specific bit in a word is set (see 2.4.8.2).

2. Subroutine LATLON. Calculate the geographic latitude and longitude of the satellite for an input time (see 2.4.8.7).

3. Subroutine OPNDOF. Open a diagnostic output file (see 2.4.8.9).

f. Interfaces.

1. Called from Subroutine MPPRA.

2. Uses COMMON blocks EPANL, LUDIAG, MPEANL, MPIANL, RPAANL, and SATEPH.

2.4.4.6 Subroutine CKLPRC.

a. Function.

Calculate an estimate of the height-integrated irregularity strength, C_kL, using high-resolution N_i data from the SM sensor or N_e data from the EP sensor (modes C, D, or DS only), and the filter PDS data from the SM sensor. A description of the theory on which this calculation is based can be found in Appendix B.5. Figure 15 is a hierarchy chart for this module.

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag
THEEND	-	End-of-data flag
IDOCKL	-	C_kL processing control flag 0: Do not calculate C_kL >1: Calculate C_kL
IDIAG	-	C_kL diagnostic print switch

```
                    ---------------------------------------IESMPEPPRT.DAT-------------------------------------
REV 00000, WX2546  I          ELECTRON           I       S/C       I  ANALYSIS  FLAGS  I
HHMMSS  GLAT GLON  I    DENSITY    I     TEMP    I   POTENTIAL     I    123456789      I
-------------------+---------------+-------------+-----------------+-------------------+
080006 -26.5 149.4 I********       I******       I*******          I    000000000      I
080010 -26.2 149.3 I********       I******       I*******          I    000000000      I
080014 -26.0 149.2 I********       I******       I*******          I    000000000      I
080018 -25.8 149.1 I********       I******       I*******          I    000000000      I
080022 -25.5 149.0 I********       I******       I*******          I    000000000      I
080026 -25.3 149.0 I********       I******       I*******          I    000000000      I
080030 -25.1 148.9 I********       I******       I*******          I    000000000      I
080034 -24.9 148.8 I********       I******       I*******          I    000000000      I
080038 -24.6 148.7 I********       I******       I*******          I    000000000      I
080042 -24.4 148.6 I********       I******       I*******          I    000000000      I

  a      b     c     d       e       f     g      h       i          j

                                    (Labels a-j cross-reference to Table 2-01.)
                  ----------------------------------------------------------------------------------------
```

Figure 14a IESMPEPPRT sample output listing

```
                ------------------------------------IESMPRPAPRT.DAT--------------------------------------
REV 00000, WX2546  I        HEAVY  ION          I        LIGHT  ION        I      RAM       I      S/C       I ANALYSIS FLAGS I
HHMMSS  GLAT GLON  I   DENSITY     I    TEMP    I   DENSITY    I    TEMP   I   DRIFT VEL    I   POTENTIAL    I 123456789   LI I
-------------------+---------------+------------+--------------+-----------+----------------+----------------+----------------+
080134 -21.3 147.9 I********  0.00 I****** 0.00 I 2.95E+4  0.00 I 9980. 5.64 I*******   0.00 I*******   0.00 I 001000000   0 I
080138 -21.1 147.8 I********  0.00 I****** 0.00 I 3.55E+4 16.04 I 5620. 2.33 I*******   0.00 I*******   0.00 I 000100000   2 I
080142 -20.9 147.8 I 2.95E+4       I  260.      I********       I******      I*******         I*******         I 000100000     I
080146 -20.7 147.7 I 1.70E+4  0.76 I  280. 0.07 I 5.75E+3  0.66 I  140. 0.04 I  -520.2298.01 I*******   0.00 I 001000000   1 I
080150 -20.4 147.6 I 2.95E+4       I 2320.      I********       I******      I*******         I -20.42         I 000000000     I
080154 -20.1 147.8 I 1.41E+4       I  600.      I 1.23E+4       I  280.      I  160.          I*******         I 001000000     I
080158 -19.9 147.7 I 2.69E+4       I 2040.      I********       I******      I*******         I -17.70         I 000000000     I
080202 -19.7 147.7 I 1.58E+4       I 1100.      I 9.33E+3       I  780.      I -360.          I*******         I 001000000     I
080206 -19.5 147.6 I 3.09E+4  1.14 I 1600. 0.81 I 5.01E+2  0.00 I****** 0.00 I  -120.-120.00 I -19.78  -1.42 I 000000000   0 I
080210 -19.2 147.5 I 3.02E+4  1.98 I  960. 0.54 I 5.01E+2  0.04 I****** 0.00 I  -800. -35.87 I -17.62   1.42 I 000000000   1 I
080214 -19.0 147.4 I 3.02E+4       I 2360.      I********       I******      I*******         I -17.64         I 000000000     I
080218 -18.8 147.4 I 4.07E+4       I  320.      I 5.01E+2       I******      I  160.          I*******         I 001000000     I
080222 -18.5 147.3 I********  0.00 I****** 0.00 I 4.90E+4  4.12 I 2840. 0.75 I*******   0.00 I -17.26   3.45 I 000000000   1 I
080226 -18.3 147.2 I********       I******      I 2.69E+4       I 5340.      I*******         I*******         I 000100000     I
080230 -18.1 147.1 I********       I******      I 2.57E+4       I 9200.      I*******         I*******         I 010100000     I
080234 -17.9 147.1 I 2.45E+4  0.90 I  600. 0.70 I********  0.00 I****** 0.00 I*******   0.00 I*******   0.00 I 000100000   0 I
080238 -17.6 147.0 I 3.02E+4       I  520.      I********       I******      I*******         I*******         I 001000000     I
080242 -17.4 146.9 I 3.09E+4  1.06 I  900. 1.64 I 5.01E+2  0.00 I****** 0.00 I  -640.-640.00 I*******   0.00 I 001000000   0 I
080246 -17.2 146.8 I********       I******      I 2.75E+4       I 2560.      I*******         I -17.62         I 000000000     I
080250 -16.9 146.8 I 2.63E+4       I  760.      I********       I******      I*******         I*******         I 000100000     I
080254 -16.6 146.9 I********  0.00 I****** 0.00 I 4.90E+4  4.06 I 3320. 0.95 I*******   0.00 I*******   0.00 I 011000000   1 I
080258 -16.4 146.8 I 2.69E+4  1.33 I  780. 0.46 I********  0.00 I****** 0.00 I*******   0.00 I*******   0.00 I 000100000   2 I
080302 -16.2 146.8 I********       I******      I 2.51E+4       I 2480.      I*******         I*******         I 001000000     I
080306 -16.0 146.7 I 3.63E+4  1.21 I 3960. 1.38 I 5.01E+2  0.00 I****** 0.00 I  760. 760.00 I*******   0.00 I 000100000   0 I
080310 -15.7 146.6 I 2.69E+4       I 3500.      I********       I******      I*******         I -16.62         I 000000000     I
080314 -15.5 146.6 I 2.09E+4  0.98 I  320. 0.29 I 1.00E+4  1.14 I  500. 0.45 I  520. 466.42 I*******   0.00 I 001000000   1 I
080318 -15.3 146.5 I 2.95E+4       I  580.      I********       I******      I*******         I -16.92         I 000000000     I
080322 -15.0 146.4 I 2.82E+4       I  800.      I********       I******      I*******         I*******         I 001000000     I
080326 -14.8 146.3 I********  0.00 I****** 0.00 I 3.24E+4  2.80 I 1060. 0.58 I*******   0.00 I*******   0.00 I 001000000   1 I
080330 -14.6 146.3 I 2.95E+4       I  440.      I********       I******      I*******         I*******         I 000100000     I
080334 -14.3 146.2 I 3.47E+4       I  600.      I********       I******      I*******         I*******         I 001000000     I
080338 -14.1 146.1 I 3.24E+4       I 2260.      I********       I******      I*******         I*******         I 011000000     I
080342 -13.9 146.1 I 2.29E+4  1.21 I 1220. 0.40 I********  0.00 I****** 0.00 I*******   0.00 I*******   0.00 I 001000000   2 I
080346 -13.7 146.0 I 2.57E+4       I  660.      I 1.74E+3       I******      I   40.          I*******         I 011000000     I
080350 -13.4 145.9 I 2.40E+4  3.19 I 1340. 0.67 I********  0.00 I****** 0.00 I*******   0.00 I -16.78   4.79 I 000000000   1 I

   a     b     c      d        e      f     g       h       i      j     k      l       m       n       o       p         q

                         (Labels a-q cross-reference to Table 2-01.)
               ----------------------------------------------------------------------------------
```

Figure 14b IESMPRPAPRT sample output listing

Table 1 Contents of MP sweep analyses comparison output files (IESMPEPPRT and IESMPRPAPRT)

EP Sweep Analyses Comparison File
(Labels cross-reference to Figure 14a)

Label	Description
a.	Time of sweep analysis (HHMMSS, GMT)
b.	Geographic latitude of sweep analysis
c.	Geographic longitude of sweep analysis
d.	Electron density from MP analysis
e.	Ratio of MP density to ground processed density
f.	Electron temperature from MP analysis
g.	Ratio of MP temperature to ground processed temperature
h.	Spacecraft potential from MP analysis
i.	MP analysis spacecraft potential minus ground processed spacecraft analysis
j.	MP analysis status/warning flags (see Volume 1, Technical Description, Appendix A: SSIES data frame word 80, cycle 1 for description)

RPA Sweep Analyses Comparison file
(Labels cross-reference to Figure 14b)

Label	Description
a.	Time of sweep analysis (HHMMSS, GMT)
b.	Geographic latitude of sweep analysis
c.	Geographic longitude of sweep analysis
d.	O+ density from MP analysis
e.	Ratio of MP density to ground processed density
f.	O+ temperature from MP analysis
g.	Ratio of MP temperature to ground processed temperature
h.	H+/He+ density from MP analysis
i.	Ratio of MP density to ground processed density
j.	H+/He+ temperature from MP analysis
k.	Ratio of MP temperature to ground processed O+ temperature
l.	Ram ion drift velocity from MP analysis
m.	MP analysis ram velocity minus ground processed ram velocity
n.	Spacecraft potential from MP analysis
o.	MP analysis spacecraft potential minus ground processed spacecraft analysis
p.	MP analysis status/warning flags (see Volume 1, Technical Description, Appendix A: SSIES data frame word 80, cycle 2 for description)
q.	Light ion flag from ground processing results 1: H+, 2: He+.

ISWCKL - Data use switch
1: Use SM density data
2: Use SM density and filter data
3: Use EP mode C or D density data

CKLPRC:	perform calculations for C_kL analysis
. BITON:	test for bit setting
. CKLSAV:	collect density and SM filter data
.. COPY:	initialize or transfer values between arrays
. PRNTON:	set PRINTAWAY status for end of processing
. COPY:	initialize or transfer values between arrays
. CKLPRP:	prepare plasma density data for C_kL calculation
.. DENFIX:	interpolate for missing density data values
.. DETRND:	detrend data and compute RMS
... COPY:	initialize or transfer values between arrays
. ENVMOD:	calculate ionospheric model irregularity parameters
.. ERF:	error function for Gaussian distribution
. BLDPDS:	calculate power spectrum from density and filter values
.. DENPDS:	compute FFT of detrended density data
... WINDOW:	calculate FFT windowing weight factors
.... COPY:	initialize or transfer values between arrays
... FFT:	interface to IEEE FFT routines
.... FR2TR:	IEEE radix 2 iteration routine
.... FR4TR:	IEEE radix 4 iteration routine
.... FORD1:	IEEE in-place reordering routine
.... FORD2:	IEEE in-place reordering routine
... BSMOO:	perform binomial weight smoothing for data
... COPY:	initialize or transfer values between arrays
.. FILPDS:	compute power spectrum from SM filter data
. TANDP:	calculate T_1 and p_1 from power spectrum
.. LSF:	perform linear least-squares fit
. PRMRNG:	check parameters against allowed ranges
. DVAVE:	compute average values of ion drift velocity
. CVEFF:	calculate effective spacecraft velocity relative to irregularities
.. CTRANS:	transform between spacecraft and geographic coordinates
.. MAGFLD:	calculate magnetic field from model
... IGRF:	initialize coefficients for IGRF 1990 model
.. CMAT:	construct transformation matrix for irregularity coordinates
. CK:	compute the C_k irregularity parameter estimate
. CKDIAG:	report C_kL processing diagnostics
.. OPNDOF:	initialize diagnostic output file
... BITON:	test for bit setting
... FILERR:	determine and report file error status
... PRNTON:	set PRINTAWAY status for end of processing
.. FILERR:	determine and report file error status

Figure 15 CKLPRC module hierarchy chart

2. COMMON blocks.

/CKLCON/

RMSLIM	-	(RMS ΔN)/N cutoff for C_kL calculation (%)
FL	-	Lowest frequency for linear PDS fit (Hz)
FU	-	Highest frequency for linear PDS fit (Hz)

MODP1 - Model p_1 use flag
0: Use only if calculated p_1 is bad
1: Use at all times
FILFAC - Filter output adjustment factor
FFTNF - Noise floor for FFT section of PDS
IWNDOW - FFT window taper flag (0-10)
IDTRND - Detrend flag
0: detrend
1: do not detrend
LSFPDS - PDS to use in log-linear fit
0: full PDS
1: decimated PDS
LOCK - Scale size for C_kL calculation (m)
ICKLU - Ion drift velocity use flag
0: use no ion drift velocity in VTOT
1: use no ram drift velocity in VTOT
2: use all ion drift velocity
CKLKP - Nominal K_p value to use in models

/DFRAME/

IDTIME - Data frame time (sec since midnight)
IQFRME - Data frame quality flag
MODEP - EP sensor mode

/EDRREC/

NPTEDR - EDR buffer pointer
EDRBUF - EDR buffer

/EPANL/

EPEDEN - EP electron density (el/cm^3)
IEPAR - EP analysis results flag

/SATEPH/

ITCKL - Seconds since midnight
IDAY - Day-of-year
GLAT - Geographic latitude (deg)
APXLAT - Apex latitude (deg)
APXLT - Apex local time (decimal hours)
VSC - Orbital velocity of satellite (m/s)
ORBINC - Inclination of satellite orbit (rad)
PHI0 - Angle from ascending node at ITCKL(1) (rad)

/SMANL/

SMIDEN - SM ion density (ion/cm^3)
SMFPDS - SM filter power spectrum ((ion/cm^3)^2/Hz)
ISMAR - SM analysis results flag

/SMCON/

SMFFRQ	-	Center frequencies of the SM filters (Hz)
SMFBW	-	Bandwidth (-6dB points) of the SM filters (Hz)
NSMFLT	-	Number of SM filters

c. Processing.

1. If a new satellite or readout REV, reset all processing flags.

2. Save the data to be used in the calculation, SM N_i or EP N_e data (Subroutine CKLSAV). If the selected data are not available, alert the user.

3. If less than 22 seconds of data have been collected, return to Subroutine PROCES.

4. If the current C_kL calculation is to be skipped, return to Subroutine PROCES.

5. Prepare the density data for the C_kL calculation as follows (Subroutine CKLPRP):

   (a) Correct for missing data where possible.

   (b) Detrend the data (quadratic trend removal).

   (c) Calculate the RMS ΔN and (RMS ΔN)/N for the center half of the data set.

6. If the data cannot be used to calculate C_kL due to missing data, or if the (RMS ΔN)/N calculated is below the user selected limit for a C_kL calculation, jump to step 15.

7. Determine the index to use for the arrays in COMMON SATEPH.

8. Calculate irregularity model parameters a, b, δ, L, and p_{1m} (Subroutine ENVMOD).

9. Construct the power density spectrum (PDS) and the decimated PDS (DPDS) (Subroutine BLDPDS).

10. Calculate T_1 and p_1 from the PDS (Subroutine TANDP).

11. If the user has specified that a model p_1 be used to calculate C_kL, set p_1 to the model value, p_{1m}.

12. If T_1 or p_1 is out-of-range, return to Subroutine PROCES without calculating C_kL.

13. Calculate the average ion drift velocities for the center 10-second period of the 21.333... second sample (512 data points) used in calculating C_kL (Subroutine DVAVE) and calculate the effective scan velocity, V_{eff} (Function CVEFF).

14. Calculate (Function CK) and error-check (Function PRMRNG) a C_kL value from T_1, p_1, and V_{eff}.

15. Generate diagnostic output if requested (Subroutine CKDIAG).

16. Return to Subroutine PROCES.

d. Outputs.

1. COMMON blocks.

/CKLANL/

DNON	-	(RMS ΔN)/N (%)
CKL	-	Height-integrated irregularity strength (C_kL)
T1	-	Value of log-linear PDS fit at f=1Hz
P1	-	Slope of log-linear PDS fit
DFREQ	-	Frequencies in decimated PDS (Hz)
DPDS	-	Decimated power density spectrum
NCKLG	-	Number of successful C_kL calculations
NCKLT	-	Number of C_kL calculations attempted
ICKLEF	-	C_kL processing error flag
ICKLAR	-	C_kL analysis results flag

e. Associated subroutines.

1. Subroutine CKLSAV. Save the high-resolution density data and the SM filter data for calculation of C_kL (see 2.4.4.6.1).

2. Subroutine CKLPRP. Prepare the high-resolution density data for calculation of C_kL (see 2.4.4.6.2).

3. Subroutine ENVMOD. Calculate model values for irregularity parameters a, b, δ, L, and p_1 (see 2.4.4.6.5).

4. Subroutine BLDPDS. Construct a power spectral density function from the high-resolution density data and the SM filter data (see 2.4.4.6.6).

5. Subroutine TANDP. Calculate T_1 and p_1 from the PDS (see 2.4.4.6.12).

6. Function PRMRNG. Determine if a parameter is within allowed range (see 2.4.8.10).

7. Subroutine DVAVE. Calculate averages of the three components of the ion drift velocity (see 2.4.4.6.14).

8. Function CVEFF. Calculate the effective scan velocity of the satellite through the ionospheric irregularities (see 2.4.4.6.15).

9. Function CK. Calculate the irregularity strength parameter, C_k (see 2.4.4.6.20).

10. Subroutine CKDIAG. C_kL processor diagnostic print routine (see 2.4.4.6.21).

f. Interfaces.

1. Called by Subroutine PROCES.

2. Uses COMMON blocks CKLANL, CKLCON, DFRAME, EDRREC, EPANL, PIDEG, SATEPH, SMANL, and SMCON.

2.4.4.6.1 Subroutine CKLSAV.

a. Function.

Save the density and SM filter data to be used in calculating C_kL.

b. Inputs.

1. Argument list.

DEN	-	Ion or electron density ($part/cm^3$)
FIL	-	SM filter data
PROFIL	-	SM filter process flag (LOGICAL)
NSMFLT	-	Number of SM filters
IDTIME	-	Data frame time (sec since midnight)
NSECS	-	Number of seconds of data saved

c. Processing.

1. If this is the start of a new satellite data set or readout REV (flagged by NSECS=0), then

   (a) If this is the start of a 22-second data-save block (any block with a time (IDTIME) ending in 4), initialize the filter buffer control words and continue to step 2.

   (b) If not, return to Subroutine CKLPRC.

2. If this is the start of a new save block, but not the start of a new satellite data set or REV, shift the density data already saved by 10 seconds (arrays DEN and MSGDEN).

3. Load the new density data (24 samples) into the save array and keep track of how many missing data flags are in the data. The missing-data count is then loaded into array MSGDEN.

4. If the SM filter data are to be saved, load them into the filter save buffer (FILDTA). If one 10-second section of the buffer is full, flip the buffer section counter (MFIL).

5. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

NSECS	-	Updated number-of-seconds count
DENDTA	-	Density data save array (22-second array)
MSGDEN	-	Density data missing-data counter array
FILDTA	-	SM filter data save buffer
MFIL	-	Filter buffer pointer (1 or 2)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.2 Subroutine CKLPRP.

a. Function.

Prepare the high-resolution ion or electron density data stored in save-array DENDTA for calculation of C_kL. This is done in two steps: (1) replace all missing data points (if possible), and (2) detrend the data.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag (LOGICAL)
DENDTA	-	Ion/electron data save array (part/cm^3)
MSGDEN	-	Missing-data counter array
IDTRND	-	Detrend flag (0: detrend, 1: no detrend)

c. Processing.

1. Replace missing data points where possible, determine if the data set has too much missing data for the C_kL calculation, and reduce the data set to a 256-word subset if necessary to construct a usable data set (Subroutine DENFIX).

2. If the data set is usable, detrend the data and calculate the RMS ΔN for the center half of the data set (Subroutine DETRND). Note that the starting address of the array to detrend is either DENDTA(9) or DENDTA(137) depending on whether the data set to use is 512 or 256 words in length. This is done to select the center 512 to 256 points from the DENDTA array for inclusion in the detrended data set.

3. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

DN	-	Detrended density data (part/cm^3)
NDN	-	Number of data points in DN (256 or 512)
RMSDN	-	RMS ΔN of the center half of DN
DNON	-	(RMS ΔN)/N (%)
GOOD	-	Data quality flag: .TRUE. - Data can be used for C_kL .FALSE. - Data cannot be used

e. Associated subroutines.

1. Subroutine DENFIX. Fill in missing data points in the density-save array and determine if too many data points are missing (see 2.4.4.6.3).

2. Subroutine DETRND. Determine and remove the quadratic trend in the density data, and calculate the RMS ΔN for the center half of the detrended data set (see 2.4.4.6.4).

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.3 Subroutine DENFIX.

a. Function.

Replace missing data points in the density save-array (linear interpolation) and determine if too many data points are missing. If a one-second block with too many missing data points (more than 8) is found, this routine will limit the data set to the center 256 data points and attempt to build a usable detrended data set from this subset.

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV flag (LOGICAL)
DENDTA	-	Density data save array (part/cm^3)
MSGDEN	-	Missing data count array

c. Processing.

1. Check each one-second (24 samples) data block as follows:

(a) If more than 6 data points are missing in a single second, then:

(1) If this second of data is located in the first or last five seconds of the data set, reset the processing pointers to process only the center 11 seconds of the data set.

(2) If it is not, set the data quality flag to -1 and jump down to step 2.

(b) If no data points are missing, jump down to step 2.

(c) Set the starting array index for this second of data (the data check starts at word 9 in the first second of data if the entire data set is to be checked, at word 137 located in the sixth second of data if only the center 11 seconds are to be checked).

(d) Check each data point for a missing-data flag. If one or two data points are missing, fill them in using a linear interpolation from the two points on either side of the one or two point data gap. If more than two points in a row are missing, set the data quality flag to -1 and jump to

step 2.

(e) Check for the end of a one-second block (the data check ends at word 520, second 22, if the entire data set is checked, or at word 392, second 16, if only the center 11 seconds are checked). If the end of the data has been reached, jump down to step 2. If the end of a second has been reached, set the missing-data counter entry for the second just checked to zero and jump back up to step 1(a).

2. Calculate the number of data points that required fixing in the entire data array (variable LSTNFX is the number of missing data points fixed in the overlapping section of density data which was fixed prior to the last C_kL calculation). If more than 25 points were filled in, set the data quality flag to -1.

3. Return to Subroutine CKLPRP.

d. Outputs.

1. Argument list.

DENDTA -	"Patched" as possible	
MSGDEN -	Zeroed where corrections made	
NDN -	Number of data points to use (256 or 512)	
NQUAL -	Data quality flag:	
	>0:	Data is good (number of corrected data points)
	-1:	Data is not good (too many missing data points)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CKLPRP.

2. Uses no COMMON blocks.

2.4.4.6.4 Subroutine DETRND.

a. Function.

Calculate and remove the quadratic trend from a data set spaced at 1/24 second intervals and calculate the RMS variation in the center half of the detrended data set. If the detrend flag is set to 1, undo the detrend by copying the raw data into the detrended data buffer. Note: The detrend is done whether or not detrended data is used in the FFT in order to calculate the RMS parameters.

b. Inputs.

1. Argument list.

DENDTA - Data to be detrended

NPTS	-	Number of data points used (256 or 512)
IDTRND	-	Detrend flag (0: detrend, 1: no detrend)

c. Processing.

1. Determine whether a 512 or 256 point fit is required.

2. Calculate the coefficients for the quadratic trend using a least squares fit.

3. Subtract the quadratic trend from the data set.

4. Calculate the RMS ΔN and (RMS ΔN)/N variation in the center half of the detrended data set.

5. If no detrend is desired, copy the original data into the detrended data array.

6. Return to Subroutine CKLPRP.

d. Outputs.

1. Argument list.

DN	-	Detrended data
RMSDN	-	RMS ΔN
DNON	-	(RMS ΔN)/N (%)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CKLPRP.

2. Uses no COMMON blocks.

2.4.4.6.5 Subroutine ENVMOD.

a. Function.

Calculate model values for ionospheric irregularity parameters a, b, δ, L, and p_1. The algorithms and model constants used are from the WBMOD ionospheric irregularity model with the exception of the model for L. The form for the model for L is taken from the model form used for the axial ratio, a.

b. Inputs.

1. Argument list.

APXLAT	-	Apex latitude (de[illegible]
APXLT	-	Apex local time (d[illegible]mal hours)
KP	-	3 hour K_p index

BNDRY	-	High latitude scintillation boundary (deg): ≤ 0.0 = boundary value will be calculated > 0.0 = boundary value to be used

2. COMMON blocks.

/SCNMOD/

AE	-	Equatorial model value for a
AA	-	Auroral model value for a
BH	-	Auroral model value for b
GMLA	-	Equatorial region boundary (deg)
GMLAW	-	Equatorial region transition width (deg)
GML1	-	Auroral boundary model lead term (deg)
CK	-	Variation of auroral boundary with K_p
CBT	-	Variation of auroral boundary with time
DBT	-	Phase of auroral boundary time variation (hours)
CHB	-	Auroral region transition width parameter
DELTA0	-	Model value for δ
LE	-	Equatorial model value for L
LM	-	Mid-latitude model value for L
LA	-	Auroral model value for L
Q	-	Model value for q

c. Processing.

1. Calculate the irregularity axial ratio parameters, a and b.

2. Calculate the location of the auroral scintillation boundary if input argument BNDRY is ≤ 0.0.

3. Set the model value for δ.

4. Calculate the value for L.

5. Calculate the model value for p_1.

6. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

A	-	Irregularity axial ratio along the magnetic field
B	-	Irregularity axial ratio across the magnetic field
DELTA	-	Angle between magnetic L-shells and irregularity sheets (rad)
LEFF	-	Effective thickness of the irregularity layer (km)
P1M	-	Model value for the PDS slope parameter, p_1.

e. Associated subroutines.

1. Function ERF. Error function (see 2.4.8.4).

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses COMMON block SCNMOD.

2.4.4.6.6 Subroutine BLDPDS.

a. Function.

Construct a power density spectrum (PDS) from the detrended high-resolution density data and the SM sensor filter data.

b. Inputs.

1. Argument list.

DN	-	Detrended density data (part/cm^3)
NDN	-	Number of points in DN (256 or 512)
FILDTA	-	SM sensor filter data
PROFIL	-	SM filter use flag (LOGICAL)
NSMFLT	-	Number of SM filters
SMFFRQ	-	Center frequencies of the SM filters (Hz)
FILFAC	-	Filter PDS adjustment factor
IWNDOW	-	FFT window parameter (0-10)
ISMOO	-	Smoother flag =0: Do not smooth the PDS ≠0: Smooth the PDS

c. Processing.

1. Construct the low-frequency portion of the spectrum from the detrended high-resolution density data (Subroutine DENPDS).

2. If the SM filter data are available, add the filter PDS data to the high-frequency end of the PDS (Subroutine FILPDS).

3. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

FREQ	-	PDS frequencies (Hz)
PDS	-	Power density spectrum ($(ion/cm^3)^2/Hz$)
NPDS	-	Number of data points in full PDS
DFREQ	-	Decimated PDS frequencies (Hz)
DPDS	-	Decimated PDS ($(ion/cm^3)^2/Hz$)

e. Associated subroutines.

1. Subroutine DENPDS. Calculate the low-frequency end of the PDS from the detrended high-resolution density data (see 2.4.4.6.7).

2. Subroutine FILPDS. Construct the high-frequency end of the PDS from the SM filter data (see 2.4.4.6.11).

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.7 Subroutine DENPDS.

a. Function.

Calculate the power density spectrum (PDS) of the input 512 point data set and a decimated PDS at equally spaced (log) frequency steps.

b. Inputs.

1. Argument list.

DN	-	Data set from which to calculate the PDS
NPTS	-	Number of data points used (256 or 512)
IWNDOW	-	FFT window parameter (0-10)
ISMOO	-	Smoother flag =0: Do not smooth the PDS ≠0: Smooth the PDS

c. Processing.

1. If the number of points has not changed, calculate the window to be used on the data (Subroutine WINDOW) and the necessary normalization factors.

2. Apply the window to the input data.

3. Perform a fast-Fourier transform on the windowed data (Subroutine FFT).

4. Convert the spectrum calculated by the FFT to a power density spectrum and calculate the frequencies of the spectrum.

5. If user-requested, smooth the PDS.

6. Decimate the full PDS to a 6 point PDS with equally spaced (log) frequencies. This is done by a simple linear interpolation.

7. Return to Subroutine BLDPDS.

d. Outputs.

1. Argument list.

FREQ	-	PDS frequencies (Hz)
PDS	-	Power density spectrum $((el/cm^3)^2/Hz)$
DFREQ	-	Frequencies in the decimated PDS (Hz)
DPDS	-	6-point PDS $((el/cm^3)^2/Hz)$

e. Associated subroutines.

1. Subroutine WINDOW. Construct the window to be applied to the input data prior to the FFT (see 2.4.4.6.8).

2. Subroutine FFT. Calculate the fast-Fourier transform of an input 512-point data set (see 2.4.4.6.9).

3. Subroutine BSMOO. Smooth the PDS (see 2.4.4.6.10).

f. Interfaces.

1. Called by Subroutine BLDPDS.

2. Uses no COMMON blocks.

2.4.4.6.8 Subroutine WINDOW.

a. Function.

Construct a simple half-lobe cosine window to be applied to a data set prior to an FFT.

b. Inputs.

1. Argument list.

NPTS	-	Number of data points in the window
IWNDOW	-	FFT window parameter (0-10): 0: do not window data >0: Use 10x(IWNDOW) % cosine taper window

2. COMMON blocks.

/PIDEG/

PI - π

c. Processing.

1. Fill the window with ones.

2. If IWNDOW is non-zero, then

(a) Calculate the percent of taper in the window.

(b) Construct the window.

3. Calculate the window normalization factor.

4. Return to Subroutine DENPDS.

d. Outputs.

1. Argument list.

W	-	Window
WNORM	-	Window normalization factor

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine DENPDS.

2. Uses COMMON block PIDEG.

2.4.4.6.9 Subroutine FFT.

a. Function.

Perform a fast-Fourier transform (FFT) on an input data set. (Note: This routine, and its associated subroutines, make up a standard FFT package developed and supported by the IEEE (See reference 1.2.3w). A copy of the description of this package from reference 1.2.3w can be found in the SSIES Project Workbook.)

b. Inputs.

1. Argument list.

B	-	Data set to transform
N	-	Number of data points in B (must be an even power of 2)

c. Processing.

1. Calculate the power of 2 and 4 for the input number of data points.

2. If the number of data points is not an even power of 4, perform the radix-2 iterations (Subroutine FR2TR).

3. Perform the radix-4 iterations (Subroutine FR4TR).

4. Perform the radix-2 (Subroutine FORD1) and radix-4 (Subroutine FORD2) reordering of the

FFT.

5. Return to Subroutine DENPDS.

d. Outputs.

1. Argument list.

B	-	FFT of the input data

e. Associated subroutines.

1. Subroutine FORD1. Perform radix-2 reordering.
2. Subroutine FORD2. Perform radix-4 reordering.
3. Subroutine FR2TR. Perform radix-2 iterations of FFT.
4. Subroutine FR4TR. Perform radix-4 iterations of FFT.

f. Interfaces.

1. Called by Subroutine DENPDS.
2. Uses no COMMON blocks.

2.4.4.6.10 Subroutine BSMOO.

a. Function.

Smooth an input data set using a centered, moving smoother with binomial weights. The smoother can be 3, 5, 7, 9, or 11 points in width. The size of the smoother is adjusted progressively downward from the user specified width to a width of 1 at the ends of the data set to avoid problems with partial smoothing at the ends.

b. Inputs.

1. Argument list.

DATA	-	Data to be smoothed
NPTS	-	Number of points in DATA
NSMOO	-	Size of smoother (1-5)

c. Processing.

1. If the size of the smoother is outside the allowed range, alert the user and set the smoother to the maximum size (5).
2. Save the first and last points (which are not smoothed in this algorithm) in the smoothed data array (SDATA).

3. Step through the remaining points, smoothing as follows:

   (a) Determine the size of the smoother based on the user-requested size and the distance (in data points) to the ends of the data set.

   (b) Calculate the smoothed value for the current data point by generating a binomial weighted sum of the data points surrounding the current point.

   (c) Store the smoothed data point in the smoothed data array (SDATA).

4. Return to Subroutine DENPDS.

d. Outputs.

   1. Argument list.

SDATA	-	Smoothed data

e. Associated subroutines.

   None.

f. Interfaces.

   1. Called by Subroutine DENPDS.

   2. Uses no COMMON blocks.

2.4.4.6.11 Subroutine FILPDS.

a. Function.

   Construct the high-frequency end of the PDS from SM sensor filter data.

b. Inputs.

   1. Argument list.

FILDTA	-	SM filter data save buffer
NSMFLT	-	Number of SM filters
SMFFRQ	-	Center frequencies of the SM filters (Hz)
FILFAC	-	Filter PDS adjustment factor

c. Processing.

   1. For each filter, sum the values in the ten one-second samples in FILDTA to construct the average PDS and add the filter samples into the PDS array with the filter frequency.

   2. Return to Subroutine BLDPDS.

d. Outputs.

1. Argument list.

FREQ	-	PDS frequencies
PDS	-	Power density spectrum from the filter data
NPDS	-	Number of values in FREQ and PDS
DFREQ	-	Frequencies of filter section of decimated PDS (Hz)
DPDS	-	Filter section of the decimated PDS

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine BLDPDS.

2. Uses no COMMON blocks.

2.4.4.6.12 Subroutine TANDP.

a. Function.

Calculate irregularity parameters T_1 and p_1 from the power density spectrum (PDS).

b. Inputs.

1. Argument list.

FREQ	-	PDS frequencies (Hz)
PDS	-	Power density spectrum ($(el/cm^3)^2/Hz$)
NFPDS	-	Number of PDS entries from the FFT calculation
FL	-	Lower frequency cutoff for log-linear fit (Hz)
FU	-	Higher frequency cutoff for log-linear fit (Hz)
FFTNF	-	Noise floor cutoff for FFT section of PDS
NPTS	-	Number of points in arrays FREQ and PDS

c. Processing.

1. On first call, validity check the user input parameters.

2. Determine the limits of the fit for either the full or decimated PDS so that the frequencies in FU and FL bound the log linear fit frequencies.

3. Convert both frequency and PDS to log.

4. If enough data points are available (50 for the full PDS, 3 for a decimated PDS), perform the log linear least squares fit (Subroutine LSF).

5. If the fit was good, set P1 and T1. Otherwise, set them to the missing data flag.

6. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

T1	-	Value of the log-linear fit at 1 Hz frequency (dB MKS)
P1	-	Slope of the log-linear fit

e. Associated subroutines.

1. Subroutine LSF. Perform a linear least squares fit to a data set (see 2.4.4.6.13).

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.13 Subroutine LSF.

a. Function.

Calculate the coefficients of a linear least squares fit to the input data set.

b. Inputs.

1. Argument list.

X	-	Array of x data points
Y	-	Array of y data points
NPTS	-	Number of points in X and Y

c. Processing.

1. Build the matrix for the linear least squares fit.

2. If the determinant of the fit matrix is non-zero, calculate the coefficients for the linear fit. Otherwise, set the fit quality flag to indicate that a fit was not found.

3. Return to Subroutine TANDP.

d. Outputs.

1. Argument list.

A0	-	Intercept of linear fit equation
A1	-	Slope of linear fit equation
GOOD	-	Good analysis flag

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine TANDP.

2. Uses no COMMON blocks.

2.4.4.6.14 Subroutine DVAVE.

a. Function.

Calculate average values for the three components of the *in situ* ion drift velocity.

b. Inputs.

1. Argument list.

ITCKL	-	Analysis time (seconds since midnight)
NPTEDR	-	EDR load buffer pointer
EDR	-	EDR load buffer (REAL)
IEDR	-	EDR load buffer (INTEGER)

c. Processing.

1. Determine which of the two EDRs stored in the EDR load buffer to use based, on the time of the current C_kL analysis and the times in the EDRs.

2. Loop through the chosen EDR buffer to calculate the average cross-track drift velocities (from DM sensor analyses) for a ten second period centered on the C_kL analysis time.

3. Select the ram drift velocity from the RPA analysis nearest in time to the C_kL analysis time.

4. Return to Subroutine CKLPRC.

d. Outputs.

1. Argument list.

UR	-	Ram drift velocity (m/s)
UH	-	Horizontal cross-track drift velocity (m/s)
UV	-	Vertical cross-track drift velocity (m/s)

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.15 Function CVEFF.

a. Function.

Calculate the effective scan velocity of the satellite through the irregularity field in the irregularity coordinate system.

b. Inputs.

1. Argument list.

VSC	-	Spacecraft orbital velocity (m/s)
UR	-	Ram ion drift velocity (m/s)
UH	-	Horizontal cross-track ion drift velocity (m/s)
UV	-	Vertical cross-track ion drift velocity (m/s)
ORBINC	-	Orbital inclination (rad)
PHI	-	Angle in orbit from ascending node (rad)
GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)
ALT	-	Altitude (km)
IYMD	-	Packed year, month, day of REV (YYMMDD)
DELTA	-	Angle between magnetic L-shells and irregularity sheets (rad)
A,B	-	Irregularity axial ratios

2. COMMON blocks.

/PIDEG/

RAD	-	Degree to radian conversion factor

c. Processing.

1. Calculate the velocity of the spacecraft with respect to the irregularities in a geographic latitude/longitude-oriented coordinate system (Subroutine CTRANS).

2. Calculate the rotation matrix required to transform the velocity into the irregularity coordinate system (Subroutine CMAT).

3. Calculate the effective velocity of the spacecraft in the irregularity coordinate system using the rotation matrix.

4. Return to Subroutine CKLPRC.

d. Outputs.

1. Function value.

CVEFF	-	Effective scan velocity (m/s)

e. Associated subroutines.

1. Subroutine CTRANS. Transform from the spacecraft coordinate system to a geographic-oriented coordinate system (see 2.4.4.6.16).

2. Subroutine MAGFLD. Calculate the vector components of the earth's magnetic field at a given latitude/longitude/altitude (see 2.4.4.6.17).

3. Subroutine CMAT. Calculate the rotation matrix to transform from a geographic-oriented coordinate system to the irregularity coordinate system (see 2.4.4.6.19).

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses COMMON block PIDEG.

2.4.4.6.16 Subroutine CTRANS.

a. Function.

Transform between a coordinate system oriented with the spacecraft motion (+x nadir, +y along orbit, +z towards sun, for an ascending daytime orbit) to a system oriented with the geographic latitude and longitude (+x north, +y east, +z nadir). A description of this transformation can be found in Section 4.1.3.2.

b. Inputs.

1. Argument list.

VFROM	-	Vector to transform
ORBINC	-	Orbital inclination (rad)
PHI	-	Angle in orbit from ascending node (rad)
GLAT	-	Geographic latitude (rad)
TOGEOG	-	Transformation direction switch (LOGICAL): .TRUE. - To geographic system .FALSE. - To spacecraft system

c. Processing.

1. Calculate the angle between the spacecraft direction and the local geographic coordinate system.

2. Rotate the input vector according to the angle just calculated and the transformation switch.

3. Return to Function CVEFF.

d. Outputs.

1. Argument list.

VTO	-	Transformed vector

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Function CVEFF.

2. Uses COMMON block PIDEG.

2.4.4.6.17 Subroutine MAGFLD.

a. Function.

Calculate the local magnetic field vectors from a real field model of the earth's magnetic field. The theory behind this routine can be found in reference 1.2.3t.

b. Inputs.

1. Argument list.

GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)
ALT	-	Altitude (km)
IYMD	-	Packed year, month, day of REV (YYMMDD)

2. COMMON blocks.

/PIDEG/

DEG	-	Degrees to radians conversion factor

c. Processing.

1. On the first call, invoke Subroutine IGRF to initialize the expansion coefficients of the magnetic field model.

2. Convert the input latitude and longitude to inverse Cartesian coordinates, and then to Cartesian coordinates, correcting for the earth's oblateness.

3. Calculate the translated expansion coefficients.

4. Use the expansion coefficients to calculate the geomagnetic field components.

4. Return to Subroutine CVEFF.

d. Outputs.

1. Argument list.

BABS	-	Total magnetic field strength (gauss)
BNORTH	-	X component of earth's magnetic field
BEAST	-	Y component of earth's magnetic field
BNADIR	-	Z component of earth's magnetic field

e. Associated subroutines.

1. Subroutine IGRF. Calculate the coefficients for a spherical harmonic expansion of the earth's magnetic field (see 2.4.4.6.18).

f. Interfaces.

1. Called by Subroutine CVEFF.

2. Uses COMMON block PIDEG.

2.4.4.6.18 Subroutine IGRF.

a. Function.

Initialize the expansion coefficients of the International Geomagnetic Reference Field. This has been updated to use the 1990 model of the earth's magnetic field. See reference 1.2.3t.

b. Inputs.

1. Argument list.

YEAR	-	year (e.g. 1993.0)

c. Processing.

1. Calculate the time variable for correcting the coefficients to the current year.

2. Sum over the components to convert them to the format required for Subroutine MAGFLD.

3. Return to Subroutine MAGFLD.

d. Outputs.

1. Argument list.

G	-	Modified field expansion coefficients

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine MAGFLD.

2. Uses no COMMON blocks.

2.4.4.6.19 Subroutine CMAT.

a. Function.

Calculate the rotation matrix used to transform from a geographic coordinate system to the irregularity coordinate system. See page 864 of reference 1.2.3t for a description of this matrix.

b. Inputs.

1. Argument list.

DIP	-	Magnetic dip angle (rad)
DELTA	-	Angle between magnetic L-shells and irregularity sheets (rad)
A,B	-	Irregularity axial ratios

c. Processing.

1. Calculate parameters used in constructing the matrix.

2. Calculate the on-diagonal terms.

3. If axial ratio B is not 1.0 (irregularity sheets), calculate the off-diagonal terms. If it is 1.0, set the off-diagonal terms to zero.

4. Return to Function CVEFF.

d. Outputs.

1. Argument list.

C	-	Transformation matrix

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Function CVEFF.

2. Uses no COMMON blocks.

2.4.4.6.20 Function CK.

a. Function.

Calculate the irregularity strength parameter C_k.

b. Inputs.

1. Argument list.

T1	-	Irregularity parameter T_1
P1	-	Irregularity parameter p_1
VEFF	-	Effective satellite velocity (m/s)

c. Processing.

1. Calculate C_k.

2. Return to Subroutine CKLPRC.

d. Outputs.

1. Function value.

CK	-	Irregularity strength parameter C_k

e. Associated subroutines.

None.

f. Interfaces.

1. Called by Subroutine CKLPRC.

2. Uses no COMMON blocks.

2.4.4.6.21 Subroutine CKDIAG.

a. Function.

Construct diagnostic output files for the C_kL analysis module.

b. Inputs.

1. Argument list.

ITCKL	-	Analysis time (seconds since midnight)
DENDTA	-	Density data used to calculate C_kL
FILDTA	-	SM filter data used to calculate C_kL
DN	-	Detrended density data used to calculate C_kL
NDN	-	Number of data points in DN

FREQ	-	PDS frequencies (Hz)
PDS	-	Power density spectrum used to calculate C_kL
NPDS	-	Number of data points in FREQ and PDS
GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)
APXLAT	-	Apex latitude (deg)
APXLON	-	Apex longitude (deg)
APXLT	-	Apex local time (hours)
A,B	-	Irregularity axial ratios
DELTA	-	Angle between L-shells and irregularity sheets (rad)
LEFF	-	Effective thickness of irregularity layer (km)
P1M	-	Model value of p_1
VEFF	-	Effective satellite scan velocity
RMSDN	-	RMS ΔN
IDIAG	-	Diagnostic print switch 0: No diagnostic output 1: Binary output file only 2: ASCII output file only 3: Both binary and ASCII output files

2. COMMON blocks.

/CKLANL/

CNON	-	(RMS ΔN)/N (%)
CKL	-	Height-integrated irregularity strength (C_kL)
T1	-	Value of log-linear PDS fit at f=1Hz
P1	-	Slope of log-linear PDS fit
DFRQ	-	Frequencies in decimated PDS (Hz)
DPDS	-	Decimated power density spectrum
ICKLEF	-	C_kL processing error flag
ICKLAR	-	C_kL analysis results flag

/CKLCON/

RMSLIM	-	(RMS ΔN)/N cutoff for C_kL calculation (%)
FL	-	Lowest frequency for linear PDS fit (Hz)
FU	-	Highest frequency for linear PDS fit (Hz)
MODP1	-	Model p_1 use flag: 0: Use only if calculated p_1 is bad 1: Use at all times
FILFAC	-	Filter output adjustment factor
FFTNF	-	Noise floor for FFT section of PDS
IWNDOW	-	FFT window taper flag (0-10)
IDTRND	-	Detrend flag: 0: detrend 1: do not detrend
LSFPDS	-	PDS to use in log-linear fit: 0: full PDS 1: decimated PDS
LOCK	-	Scale size for C_kL calculation (m)

ICKLU	-	Ion drift velocity use flag 0: use no ion drift velocity in VTOT 1: use no ram drift velocity in VTOT 2: use all ion drift velocity
CKLKP	-	Nominal K_p value to use in models

/LUDIAG/

LUCKLA	-	Logical unit number of ASCII output file
LUCKLB	-	Logical unit number for binary output file

/PROCON/

IFLT	-	Satellite flight ID

c. Processing.

1. On first call to this routine, open the necessary ASCII and/or binary output files (Subroutine OPNDOF).

2. If neither file could be opened, set the diagnostic print flag to zero (disabling the diagnostic output) and return to Subroutine CKLPRC.

3. If an ASCII output file is requested, then

   (a) If a RESET=.TRUE. call, output the sensor processing constants and processing control information for the CKL module.

   (b) If the data frame was good, write out all data input and produced in the current invocation of the CKL processor.

4. If a binary output file is requested, then, if the frame was good, write the diagnostic output buffer to the output file.

5. Return to Subroutine CKLPRC.

d. Outputs.

IESCKLPRT. ASCII diagnostic output file (see Section 4.4.5).

IESCKLDIAG. Binary diagnostic output file (see Section 4.4.5).

e. Associated subroutines.

1. Subroutine OPNDOF. Open diagnostic output files (see 2.4.8.9).

2. Subroutine FILERR. File error handling routine (see 2.4.8.5).

f. Interfaces.

1. Called from Subroutine CKLPRC.

2. Uses COMMON blocks CKLANL, CKLCON, LUDIAG, and PROCON.

2.4.4.7 Subroutine QCPRC.

a. Function.

Collect parameters stored in file IESSTATFILE to include REV averages of various parameters generated from the analysis of data from the SSIES sensors, REV averages of the difference in plasma density calculated by the various sensors, analysis success counters, and the data frame skip counter. These are all loaded into a buffer which is output to file IESSTATFILE when the end of data is reached for a satellite or readout REV. Figure 16 is a hierarchy chart for this module.

QCPRC:	perform and report comparisons of processor results
. SCDIAG:	report spacecraft and processing diagnostics
. . BITON:	test for bit setting
. IAVRGE:	compute an integer average for specified sum
. AVRAGE:	compute a floating point average for specified sum
. LDSTF:	prepare IESSTATFILE for new status report
. . TIMCON:	convert between UT and IES reference minutes
. . PRNTON:	set PRINTAWAY status for conclusion of processing
. . COPY:	initialize array or transfer values between arrays
. . SFSUM:	report summary records
. . FILERR:	determine and report file error status
. COPY:	initialize array or transfer values between arrays
. QCRPA:	obtain ion data from RPAPRC or MPPRC results
. QCEP:	obtain electron data from EPPRC or MPPRC results

Figure 16 QCPRC module hierarchy chart

b. Inputs.

1. Argument list.

THEEND	-	End-of-data flag

2. COMMON blocks.

/CKLANL/

CKL	-	Integrated irregularity spectral strength
NCKLG	-	Number of good C_kL analyses in current REV
NCKLT	-	Number of total analyses attempted
ICKLEF	-	C_kL analysis error flag
ICKLAR	-	C_kL analysis results flag

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
IQFRME	-	Data frame quality flag
MODEP	-	EP sensor mode (0-6: A-E)

/DMANL/

DMAUH	-	Average horizontal ion drift velocity (m/s)
DMAUV	-	Average vertical ion drift velocity (m/s)
DMIDEN	-	Ion density (ion/cm^3)
IDMEF	-	DM analysis error flag
IDMAR	-	DM analysis results flag

/EPANL/

IEPTIM	-	Sweep center time (seconds since midnight)
EPEDEN	-	DC mode electron density (24, el/cm^3)
EPADEN	-	(DC mode) Average electron density (el/cm^3) (sweep mode) Electron density (el/cm^3)
EPETMP	-	Electron temperature (°K)
EPVPOT	-	EP sensor potential (volts)
NEPSG	-	Number of successful sweep analyses
NEPST	-	Number of sweep analyses attempted
IEPEF	-	EP analysis error flag
IEPAR	-	EP analysis results flag

/MPEANL/

METIME	-	EP sweep center time (seconds since midnight)
EDENMP	-	MP EP electron density (el/cm^3)
ETMP	-	MP EP electron temperature (°K)
SPEPMP	-	MP EP satellite potential (volts)
IMPEAR	-	MP EP sweep analysis results flag

/MPIANL/

MITIME	-	RPA sweep center time (seconds since midnight)
ODENMP	-	MP RPA O+ density (ion/cm^3)
OTMP	-	MP RPA O+ temperature (°K)
HDENMP	-	MP RPA light ion density (ion/cm^3)
HTMP	-	MP RPA light ion temperature (°K)
URMP	-	MP RPA ram ion drift velocity (m/s)
SPRPMP	-	MP RPA satellite potential (volts)
IMPIAR	-	MP RPA analysis results flag

/PROCON/

IFLT	-	Satellite flight ID
MISSID	-	Satellite mission ID
NUMREV	-	Readout REV number
IDOQC	-	QCPRC module processing control flag
IDBEP	-	EP analysis source for database
IDBRP	-	RPA analysis source for database
ISWCKL	-	Data source for C_kL analysis
IRMIN	-	Time of current run (IES minutes)
NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag

/RPAANL/

RPHDEN	-	Light ion density (ion/cm^3)
RPODEN	-	O+ ion density (ion/cm^3)
RPEDEN	-	Total ion density (ion/cm^3)
RPITMP	-	Ion temperature (°K)
RPAUR	-	Ram ion drift velocity (m/s)
RPVPOT	-	Satellite potential (volts)
NRPASG	-	Number of successful RPA sweep analyses
NRPAST	-	Number of RPA sweeps analyses attempted
IRPEF	-	RPA analysis error flag
IRPAR	-	RPA analysis results flag

/SATEPH/

IYMD	-	Dates for EDRs (6) (YYMMDD)

/SMANL/

SMADEN	-	Average ion density (ion/cm^3)
SMFPDS	-	Irregularity PDS estimates (9) ((ion/cm^3)^2/Hz)
ISMEF	-	SM analysis error flag
ISMAR	-	SM analysis results flag

/SCSOH/

AELTMP	-	Running average SSIES electronics temperature
ADSMTP	-	Running average DM/SM electronics temperature
AADCTP	-	Running average ADC temperature
ADMV0	-	Running average DM sensor offset voltage

c. Processing.

1. If requested, output spacecraft/processing status information (Subroutine SCDIAG).

2. If a new satellite, new REV, or end of data is encountered, then:

   (a) If this is not the first call to this routine, calculate the REV averages of the data in the save buffer, load various data into the buffer for IESSTATFILE, and write the buffer out to the file (Subroutine LDSTF).

   (b) Initialize all save variables and arrays.

3. Collect the following data from the current one-second data analyses when available:

   (a) C_kL analyses
   (b) Increment missing frame counter as necessary
   (c) REV date/time/number (on first time through)
   (d) SM averages (density and first filter) and running mean of density
   (e) DM averages (density and velocities) and running mean of density
   (f) RPA averages (densities, temperature, velocity) (from either RPAPRC or MPPRC)
   (g) EP averages (density, temperature) (from either EPPRC or MPPRC)

(h) Satellite potential information

4. As data are available, add to the following comparisons:

   (a) SM density to DM, EP, and RPA density
   (b) DM density to EP and RPA density
   (c) EP density to RPA density
   (d) Compare T_e to T_i (should have $T_e \geq T_i$)

5. Return to Subroutine PROCES.

d. Outputs.

   1. Argument list.

      ISTAT - Return status

e. Associated subroutines.

   1. Subroutine SCDIAG. Print out diagnostic messages concerning changes in sensor status or in processing (see 2.4.4.7.1).

   2. Function AVRAGE. Calculate the average of a real number. If the input count is zero, set the average to -1.0E37.

   3. Function IAVRGE. Calculate the average of an integer number. If the input count is zero, set the average to -999999.

   4. Subroutine LDSTF. Output the statistics buffer to file IESSTATFILE (see 2.4.4.7.2).

   5. Subroutine QCRPA. Extract data for an RPA sweep (see 2.4.4.7.3).

   6. Subroutine QCEP. Extract data for an EP sweep (see 2.4.4.7.4).

f. Interfaces.

   1. Called from Subroutine PROCES.

   2. Uses COMMON blocks CKLANL, DFRAME, DMANL, EPANL, MPEANL, MPIANL, PROCON, RPAANL, SATEPH, SMANL, and SCSOH.

2.4.4.7.1 Subroutine SCDIAG.

a. Function.

   Produce diagnostic print for following the status of the SSIES sensors and changes in processing status.

b. Inputs.

1. Argument list.

THEEND	-	End-of-data flag

2. COMMON blocks.

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
ICYCLE	-	Data frame cycle ID (1 or 2)
ICCNTR	-	Sweep program counter
IQFRME	-	Data frame quality flag
MODEP	-	EP sensor mode (0-6: A-E)
MONOLS	-	OLS command from last cycle 1 data frame
MONDSM	-	DSM command from last cycle 2 data frame

/PROCON/

IFLT	-	Satellite flight ID
NUMREV	-	Readout REV number
NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag
GRESET	-	General reset flag

c. Processing.

1. Generate diagnostics for a new satellite/new REV/general reset/end-of-data condition.

2. If a new satellite or REV, initialize all counters. Otherwise, increment all counters.

3. Alert the user for any of the following:

   (a) Duplicate/missing frames
   (b) Time jumps
   (c) Sweep clock jumps
   (d) Calibration sweeps
   (e) OLS command changes
   (f) EP mode changes
   (g) DM and SM sensor command changes

4. Return to Subroutine QCPRC.

d. Outputs.

This routine produces one line diagnostic messages described in the Programmer's Guide.

e. Associated subroutines.

1. Function BITON. Determine if a particular bit in a word is set (see 2.4.8.2).

f. Interfaces.

1. Called from Subroutine QCPRC.

2. Uses COMMON blocks DFRAME and PROCON.

2.4.4.7.2 Subroutine LDSTF.

a. Function.

Load the REV average statistics buffer built by QCPRC into file IESSTATFILE.

b. Inputs.

1. Argument list.

MSID	-	Satellite mission ID
IFLT	-	Satellite flight ID
IRMIN	-	Run wall time (IES minutes)
ISBUF	-	Buffer containing the statistics for output

2. COMMON blocks.

/IONUM/

LUSTAT	-	Logical unit for file IESSTATFILE

/STATF/

MSIDSF	-	Satellite mission ID
IFLTSF	-	Satellite flight ID
LTMUPT	-	Last time this section updated (IES Minutes)
LTMDIS	-	Last time this section listed (IES Minutes)
NREVS	-	Number of readout REVs stored in this section
ISFBUF	-	Summary statistics for up to 18 REVs

c. Processing.

1. On first call, build a directory of the satellites on file IESSTATFILE and their location.

2. Determine which of the three locations in the file corresponds to the satellite for the current data.

3. If no location is currently assigned for the satellite (new satellite, hopefully), either initialize the next empty location, or, if none are empty, reinitialize the oldest section if no data have been loaded into that section in the past 10 days. If no location is available (highly unlikely), alert the user and drop the summary buffer into the bit bucket.

4. If a location was found, read in the entire 18 REV record for that location, push down the array, and load in the new buffer.

5. Write the buffer back out to the file.

6. Build an HSP summary of the buffer just saved (Subroutine SFSUM).

7. Return to Subroutine QCPRC.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. Files.

IESSTATFILE. File to which a new summary buffer is added.

e. Associated subroutines.

1. Subroutine FILERR. File error contingency routine (see 2.4.8.5).

2. Subroutine PRNTON. Turn on the PRINTAWAY disable switch and set the console error flag (see 2.4.8.11).

3. Subroutine TIMCON. Convert to/from IES minutes (see 2.4.8.12).

4. Subroutine SFSUM. Build an HSP output summary of a summary record from file IESSTATFILE (see 2.4.6.1).

f. Interfaces.

1. Called-from Subroutine QCPRC.

2. Uses COMMON blocks IONUM and STATF.

2.4.4.7.3 Subroutine QCRPA.

a. Function.

Select the data required for Subroutine QCPRC from an RPA sweep analysis from either the RPAPRC or MPPRC modules.

b. Inputs.

1. Argument list.

ODEN	-	O+ density
HDEN	-	Light ion density
TI	-	Ion temperature
UR	-	Ion ram drift velocity
IADRP	-	Average total ion density
NDRP	-	Number of densities in average
IATI	-	Average ion temperature
NTI	-	Number of temperatures in average

IAUR	-	Average ion ram drift velocity
NUR	-	Number of velocities in average

c. Processing.

1. Construct the total ion density from the input sweep analyses.

2. If the ion density, temperature, and ram drift velocity are available, add them to their respective sums, set the output values to the input values for each, and set the availability flags.

3. Return to Subroutine QCPRC.

d. Outputs.

1. Argument list.

RPDEN	-	Total ion density
IADRP	-	Average total ion density
NDRP	-	Number of densities in average
GOTDRP	-	Total ion density available flag
RPTI	-	Ion temperature
IATI	-	Average ion temperature
NTI	-	Number of temperatures in average
GOTTI	-	Ion temperature available flag
IAUR	-	Average ion ram drift velocity
NUR	-	Number of velocities in average

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine QCPRC.

2. Uses no COMMON blocks.

2.4.4.7.4 Subroutine QCEP.

a. Function.

Select the data required for Subroutine QCPRC from an EP sweep analysis from either the EPPRC or MPPRC modules.

b. Inputs.

1. Argument list.

IEPAR	-	Analysis results flag
AEDEN	-	Average electron density (from DC mode)
EDEN	-	Electron density (from sweep)

ETMP	-	Electron temperature
SCPOT	-	Satellite potential
IADEP	-	REV average electron density
NDEP	-	Number of densities in average
IATE	-	REV average electron temperature
NTE	-	Number of temperatures in average
AVEEPV	-	REV average EP analysis satellite potential
NEPV	-	Number of potentials in average

c. Processing.

1. If DC mode data are available:

(a) Add the average density to the REV averaged sum of the density.

(b) Set the output EP density to the average density.

(c) Set the data availability flags.

2. If a sweep analysis is available:

(a) Add electron density, temperature, and satellite potential to the REV averages if they are available.

(b) Set the output EP density, temperature, and potential to the input values.

(c) Set the data availability flags.

d. Outputs.

1. Argument list.

EPDEN	-	EP electron density
IADEP	-	REV average electron density
NDEP	-	Number of densities in average
GOTDEP	-	EP density available flag
EPTE	-	EP electron temperature
IATE	-	REV average electron temperature
NTE	-	Number of temperatures in average
GOTTE	-	EP temperature available flag
AVEEPV	-	REV average EP analysis satellite potential
NEPV	-	Number of potentials in average
EPSWP	-	Sweep data flag

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine QCPRC.

2. Uses no COMMON blocks.

2.4.5 Subroutine OUTPUT.

a. Function.

Control the construction and output of SSIES EDRs to file IESEDRFILE and of AGDB summary records to a transfer file (IESAGDBXFR1 or IESAGDBXFR2). Figure 17 is a hierarchy chart for this module.

OUTPUT:	generate IESEDRFILE and IESAGDBXFR records
. WRTEDR:	write EDRs to output
.. TIMCON:	convert between UT and IES reference minutes
.. PRNTON:	set PRINTAWAY status for end of processing
.. COPY:	initialize or transfer values between arrays
.. EDRPRT:	generate an EDR summary text report
... OPNDOF:	initialize diagnostic output file
.... BITON:	test for bit setting
.... FILERR:	determine and report file error status
.... PRNTON:	set PRINTAWAY status for end of processing
.. FILERR:	determine and report file error status
. WRTXFR:	process AGDB summary records
.. LDXFR:	generate AGDB record from EDR
.. XFRPRT:	generate an AGDB record text report
... OPNDOF:	initialize diagnostic output file
.... BITON:	test for bit setting
.... FILERR:	determine and report file error status
.... PRNTON:	set PRINTAWAY status for end of processing
.. COPY:	initialize or transfer values between arrays
.. PRNTON:	set PRINTAWAY status for end of processing
.. FILERR:	determine and report file error status
. SETEDR:	initialize EDR segments for IESEDRFILE
.. PRNTON:	set PRINTAWAY status for end of processing
.. TIMCON:	convert between UT and IES reference minutes
.. COPY:	initialize or transfer values between arrays
.. FILERR:	determine and report file error status
. LDEDR:	store processed data into EDR output arrays
.. COPY:	initialize or transfer values between arrays
.. LDSWPS:	acquire EP or RPA results for EDR

Figure 17 OUTPUT module hierarchy chart

b. Inputs.

1. Argument list.

THEEND - End-of-data flag (LOGICAL)

2. COMMON blocks.

/DFRAME/

IQFRME	-	Data frame quality flag

/EDRREC/

NPTEDR	-	EDR load buffer pointer
EDR	-	EDR load buffer

/PROCON/

IFLT	-	Satellite flight ID
IEDDGP	-	EDR diagnostic print flag
IAGDGP	-	AGDB summary record diagnostic print flag
NEWSAT	-	New satellite flag
NEWREV	-	New readout REV flag

/SATEPH/

ITEDR	-	EDR times from ephemeris (seconds since midnight)
IYMD	-	EDR dates from ephemeris (YYMMDD)

c. Processing.

1. Set local reset flags based on the new satellite, new REV, and end-of-data flags. If the current data frame was bad, return to the main routine.

2. If the local reset flag is set, then:

   (a) If this is not the first time through this routine, output the current EDR I/O buffer (Subroutine WRTEDR) and build and output an AGDB summary record (WRTXFR). If there are no more data, return to the main routine. If any errors were encountered, set the return status to -1 and return to the main routine.

   (b) Set up a new EDR load buffer and EDR I/O buffer for the new satellite or REV (Subroutine SETEDR). If any errors were encountered, set the return status to -1 and return to the main routine.

3. Load the data-processed from the current data frame into the EDR load buffer (Subroutine LDEDR).

4. If an EDR has been completed and is ready for output (based on variable DONE returned from Subroutine LDEDR), output the EDR (Subroutine WRTEDR), and build and output an AGDB summary record (Subroutine WRTXFR). If any errors were encountered, set the return status to -1 and return to the main routine.

5. Set the return status to zero and return to the main routine.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

e. Associated subroutines.

1. Subroutine WRTEDR. Control output of EDRs to file IESEDRFILE (see 2.4.5.1).

2. Subroutine WRTXFR. Control construction and output of AGDB summary records to the transfer file (see 2.4.5.2).

3. Subroutine SETEDR. Initialize the EDR load and I/O buffers (see 2.4.5.3).

4. Subroutine LDEDR. Load data from the analysis COMMON blocks into the EDR load buffer (see 2.4.5.4).

f. Interfaces.

1. Called from the main routine.

2. Uses COMMON blocks DFRAME, EDRREC, PROCON, and SATEPH.

2.4.5.1 Subroutine WRTEDR.

a. Function.

Control EDR output to file IESEDRFILE.

b. Inputs.

1. Argument list.

RESET	-	Local reset flag (see OUTPUT)
EDR	-	EDR to be output (REAL)
IEDR	-	EDR to be output (INTEGER, Equivalenced to EDR)
IDIAG	-	EDR diagnostic print flag

2. COMMON blocks.

/EDRWRK/

NHREC	-	File record number for current satellite header
IHEDR	-	File header for current satellite
IBPFLG	-	EDR I/O buffer processing status flags
NREC	-	File record number for current EDR I/O record
IOEDR	-	Current EDR I/O record

/IONUM/

LUEDR - Logical unit for file IESEDRFILE

/CFINFO/

IENREC - Number of I/O records per satellite in IESEDRFILE

3. Files.

EDR I/O buffers are input from file IESEDRFILE.

c. Processing.

1. Determine the location of the EDR I/O record in which the new EDR should be stored. If the location is negative, indicating that the EDR is for a time previous to the oldest data in the file, alert the user and return to Subroutine OUTPUT without storing the EDR in file IESEDRFILE.

2. If the time in the EDR leads to an erroneous file address, indicating a possible bad time in the EDR, alert the user and return to Subroutine OUTPUT without storing the EDR in file IESEDRFILE.

3. If the new EDR does not go in the EDR I/O buffer currently in COMMON EDRWRK, then write the current I/O buffer out to file IESEDRFILE and reinitialize the I/O buffer:

   (a) If the new EDR belongs in an I/O buffer which is already in the file and has data in it, read in that I/O buffer, or
   (b) Set up a new I/O buffer by zeroing out the six processing words located in the first six locations in the EDR I/O buffer.

4. Copy the new EDR into the proper location in the I/O buffer.

5. If this is the last EDR for the current satellite/REV, then:

   (a) output the current EDR I/O buffer,
   (b) update the oldest/latest data flags in the header block,
   (c) output the header block.

6. If requested by the user, write out an ASCII list of the EDR just processed (Subroutine EDRPRT).

7. Set the return status to zero and return to Subroutine OUTPUT.

d. Outputs.

1. Argument list.

ISTAT - Return status

2. COMMON blocks.

/EDRWRK/

IHEDR	-	File header for current satellite
IBPFLG	-	EDR I/O buffer processing status flags
NREC	-	File record number for current EDR I/O record
IOEDR	-	Current EDR I/O record

3. Files.

EDR I/O buffers and the header block are output to file IESEDRFILE.

e. Associated subroutines.

1. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

2. Subroutine TIMCON. Convert between date/time and IES minutes (see 2.4.8.12).

3. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

4. Subroutine EDRPRT. Write an ASCII-format list of an EDR to a diagnostic output file (see 2.4.5.1.1).

f. Interfaces.

1. Called from Subroutine OUTPUT.

2. Uses COMMON blocks EDRWRK, IONUM, and CFINFO.

2.4.5.1.1 Subroutine EDRPRT.

a. Function.

Build and output an ASCII-format EDR summary print.

b. Inputs.

1. Argument list.

IDIAG	-	Diagnostic print flag
EDR	-	EDR to display (REAL)
IEDR	-	EDR to display (INTEGER, Equivalenced to EDR)
NRS	-	Record number for current EDR I/O buffer
NRW	-	Record number for last output EDR I/O buffer
NHREC	-	Record number of EDR file header information
NEDR	-	EDR work array pointer
IHEDR	-	File header for current satellite
IWRT	-	EDR I/O output flag
FINAL	-	Last EDR flag

2. COMMON blocks.

/LUDIAG/

LUEDRA	-	Logical unit for EDR diagnostic output file

/CKLANL/

DFREQ	-	Frequencies in decimated PDS (Hertz)

c. Processing.

1. On first call to this routine, open the diagnostic output file (Subroutine OPNDOF). If this fails, set the diagnostic print flag to zero so that no further attempts will be made to output EDR to a diagnostic print file.

2. Write out header information for the EDR I/O buffer.

3. If the current EDR is not to be listed (controlled by the value of IDIAG), return to Subroutine WRTEDR.

4. Write out the EDR.

5. Return to Subroutine WRTEDR.

d. Outputs.

Figure 30 (section 4.4.5) illustrates an EDR diagnostic output.

1. Argument list.

IDIAG	-	Set to zero if the output file is bad

2. COMMON Blocks.

None.

e. Associated routines.

1. Subroutine OPNDOF. Open a diagnostic output file (see 2.4.8.9).

f. Interfaces.

1. Called from Subroutine WRTEDR.

2. Uses COMMON blocks LUDIAG and CKLANL.

2.4.5.2 Subroutine WRTXFR.

a. Function.

Build and output EDR summary records.

NOTE: The transfer file is a remnant of GWC processing and could probably be eliminated for SFC purposes. (See Subroutine OPNXFR 2.4.2.7, for related comments.)

b. Inputs.

1. Argument list.

RESET	-	Local reset flag (see OUTPUT)
EDR	-	EDR to summarize (REAL)
IEDR	-	EDR to summarize (INTEGER, Equivalenced to EDR)
FINAL	-	Last EDR flag
IDIAG	-	Diagnostic output flag

2. COMMON blocks.

/IONUM/

LUXFR	-	Logical unit for file IESAGDBXFRn

/LUDIAG/

LUXFRA	-	Logical unit for diagnostic output

/PROCON/

IFLT	-	Satellite flight ID
NFXFR	-	Transfer file number (1 or 2)
IRDATE	-	Current run date (YYMMDD)
IRTIME	-	Current run time (HHMMSS)

3. Files.

The header buffer is input from file IESAGDBXFRn.

c. Processing

1. On first call to this routine, set up the header buffer in the first 22 words of the first I/O buffer.

2. If the transfer file is not yet full, then:

   (a) Extract a summary record from the current EDR and load it into the I/O buffer (Subroutine LDXFR),

   (b) If requested by the user, write a copy of the summary record to an ASCII diagnostic file (Subroutine XFRPRT),

(c) If the I/O buffer is full, write it out to the transfer file selected in Subroutine OPNXFR,

(d) Increment the report counter in the header array.

3. If the transfer file is full, extract a summary record from the current EDR, write out an ASCII print of the summary record, and increment the lost records counter (LOST).

4. If this is the final EDR, then:

(a) Output the current I/O buffer,

(b) Input the header record, update it as necessary, and write it back to the file,

(c) If there were lost records due to file overflow, alert the user,

(d) Return to Subroutine OUTPUT.

5. If the next EDR will be for a new satellite, update the header to show the start of data from a new satellite.

6. Return to Subroutine OUTPUT.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. Files.

EDR summary records are written to the transfer file (IESAGDBXFR1 or IESAGDBXFR2).

e. Associated subroutines.

1. Subroutine LDXFR. Extract a summary record for the AGDB from an EDR (see 2.4.5.2.1).

2. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

3. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

4. Subroutine XFRPRT. Write an ASCII format list of an EDR summary record to a diagnostic output file (see 2.4.5.2.2).

f. Interfaces.

1. Called from Subroutine OUTPUT.

2. Uses COMMON blocks IONUM and PROCON.

2.4.5.2.1 Subroutine LDXFR.

a. Function.

Build EDR summary records for the AGDB.

b. Inputs.

1. Argument list.

EDR	-	EDR to summarize (REAL)
IEDR	-	EDR to summarize (INTEGER, Equivalenced to EDR)

c. Processing.

1. Load satellite flight ID, date, time, geographic latitude, and geographic longitude from the EDR into the summary record.

2. Calculate and load the following averages:

(a) plasma density,
(b) ion ratio (ratio of O+ density to light ion density),
(c) ion temperature, and
(d) electron temperature.

3. Extract and load the C_kL, p_1, and analysis information for each C_kL analysis.

4. Return to Subroutine WRTXFR.

d. Outputs.

1. Argument list.

AGBUF	-	Summary record buffer (REAL)
IAGBUF	-	Summary record buffer (INTEGER, Equivalenced to AGBUF)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine WRTXFR.

2. Uses no COMMON blocks.

2.4.5.2.2 Subroutine XFRPRT.

a. Function.

Write an AGDB summary record to an ASCII format diagnostic output file.

b. Inputs.

1. Argument list.

AGBUF	-	Summary record buffer (REAL)
IAGBUF	-	Summary record buffer (INTEGER, Equivalenced to AGBUF)
IDIAG	-	Diagnostic print flag

2. COMMON blocks.

/LUDIAG/

LUXFRA	-	Logical unit for diagnostic output file

c. Processing.

1. On first call to this routine, open the diagnostic output file (Subroutine OPNDOF). If this fails, set the diagnostic print flag to zero so that no further attempts will be made to output EDR to a diagnostic print file.

2. Write out the ASCII format record.

3. Return to Subroutine WRTXFR.

d. Outputs.

Figure 33 (section 4.4.5) illustrates the format of a single AGDB summary record as output by this routine.

1. Argument list.

IDIAG	-	Set to zero if the output file is bad

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine WRTXFR.

2. Uses COMMON block LUDIAG.

2.4.5.3 Subroutine SETEDR.

a. Function.

Interface with file IESEDRFILE to initialize EDR I/O buffers at the start of processing data from either a new satellite or a new readout REV.

NOTE: Most (if not all) of the logic in this routine is based upon requirements for the AFGWC processing system and can be eliminated for production processing at AFSFC. The salient points to consider are:

1. The GWC input stream could (and did) contain data for multiple revs of multiple satellites. The EDR output processing was designed to segregate the processing results from each satellite into its own database storage area. The SFC input stream contains one rev of data for one satellite.

2. The GWC input stream data was not properly time ordered. The EDR output processing was designed to write the processing results to the database file in time ascending order. The SFC input stream data begins at the start time of the rev and progresses in correct time order to the end of the rev.

See Subroutine OPNEDR 2.4.2.5 for related comments.

b. Inputs.

1. Argument list.

NEWSAT	-	New satellite flag
IFLT	-	Satellite flight ID
IDATE	-	Date of EDR required for current data frame (YYMMDD)
ITIME	-	Time of EDR required for current data frame (HHMM)

2. COMMON blocks.

/CFINFO/

IENREC	-	Number of I/O records per satellite in IESEDRFILE

/IONUM/

LUEDR	-	Logical unit for IESEDRFILE

3. Files.

(a) IESEDRFILE. The header block for the satellite specified by variable IFLT and the EDR I/O block (if it exists) specified by variables IDATE and ITIME are input.

c. Processing.

1. If the initialization is for a new satellite (or at the start of a program run):

(a) Search through the EDR file header blocks for the header block for satellite IFLT and input

it. If none is found, alert the user (Subroutine PRNTON), set the return status to -1, and return to Subroutine OUTPUT.

(b) If the file has been reinitialized, set up the header block for satellite IFLT based on date IDATE and time ITIME.

2. Determine where in the file data from the current REV should be placed, and set up the EDR load and I/O buffers as follows:

(a) If the data are from a time prior to the oldest time in the EDR file, write out a warning message. If the data are over 120 minutes old, flag this as an error. Continue processing by filling the EDR load buffer with missing data flags (-1.0E37), zeroing the EDR I/O buffer processing status flags (IOEDR), and setting the EDR load buffer pointer (NPTEDR) to -1.

(b) If the data time is between the oldest and latest times in the EDR file, input the EDR I/O record which includes the current date (IDATE) and time (ITIME). If a valid EDR for the current date and time is found in the EDR I/O buffer, it is copied into the EDR load buffer and the EDR load buffer pointer (NPTEDR) is initialized to 1. If no EDR is found, initialize the EDR load buffer by filling it with missing data flags and set NPTEDR to -1.

(c) If the data time is beyond the time in the EDR file, the file is brought up to the current date/time by zeroing all processing flags in the header (IBPFLG) corresponding to EDR I/O buffers between the latest time in the file and the time of the current data. Initialize the EDR load buffer by filling it with missing data flags and set NPTEDR to -1, and zero out the processing status flags in the EDR I/O buffer (IOEDR).

3. Set the return status to zero, and return to Subroutine OUTPUT.

d. Outputs.

1. Argument list.

ISTAT	-	Return status

2. COMMON blocks.

/EDRREC/

NPTEDR	-	EDR load buffer pointer
EDR	-	EDR load buffer

/EDRWRK/

NHREC	-	File record number for current satellite header
IHEDR	-	File header for current satellite
IBPFLG	-	EDR I/O buffer processing status flags
NREC	-	File record number for current EDR I/O record
IOEDR	-	Current EDR I/O record

e. Associated subroutines.

1. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

2. Subroutine TIMCON. Convert between date/time and IES minutes (see 2.4.8.12).

3. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

f. Interfaces.

1. Called from Subroutine OUTPUT.

2. Uses COMMON blocks CFINFO, EDRREC, EDRWRK, and IONUM.

2.4.5.4 Subroutine LDEDR

a. Function.

Load the analysis results from the six processor routines (EPPRC, RPAPRC, MPPRC, DMPRC, SMPRC, and CKLPRC) into an Environmental Data Record (EDR).

b. Inputs.

1. Argument list.

RESET	-	New satellite/REV switch (LOGICAL)
GONE	-	Indicator that an EDR has been output

2. COMMON blocks.

/CKLANL/

DNON	-	(RMS ΔN)/N (%)
CKL	-	Height-integrated irregularity strength (C_kL)
T1	-	Value of log-linear PDS fit at f=1Hz
P1	-	Slope of log-linear PDS fit
DPDS	-	Decimated power density spectrum
ICKLAR	-	C_kL analysis results flag

/DFRAME/

IDTIME	-	Time of current data frame (seconds since midnight)
IQFRME	-	Bad frame flag
MODEP	-	EP sensor mode (0-6: A-E)
VBIAS	-	Ion sensor bias voltage (volts)

/DMANL/

DMAUH	-	Average horizontal, cross-track drift velocity (meters/sec)
DMAUV	-	Average vertical, cross-track ion drift velocity (meters/sec)
DMIDEN	-	Ion density (ion/cm^3)
IDMAR	-	DM analysis results flag

/EDRREC/

NPTEDR	-	EDR load buffer pointer

/EPANL/

IEPTME	-	Center time of EP sweep (sec since midnight)
EPEDEN	-	EP DC mode electron densities (el/cm^3)
EPADEN	-	EP average electron density (el/cm^3)
EPETMP	-	EP electron temperature (°K)
EPVPOT	-	EP analysis of spacecraft potential (volts)
IEPAR	-	EP analysis results flag

/MPEANL/

METIME	-	Center time of EP sweep (sec since midnight)
EDENMP	-	MP electron density (el/cm^3)
ETMP	-	MP electron temperature (°K)
SPEPMP	-	MP EP analysis of spacecraft potential (volts)
MPEFLG	-	MP EP analysis flags
IMPEAR	-	MP EP analysis results flag

/MPIANL/

MITIME	-	Center time of RPA Sweep (sec since midnight)
OOENMP	-	MP O+ ion density (ion/cm^3)
OTMP	-	MP O+ ion temperature (°K)
HDENMP	-	MP H+ ion density (ion/cm^3)
IMPLIF	-	MP light ion flag
URMP	-	MP ram ion velocity (meters/sec)
MPIFLG	-	MP RPA analysis flags
IMPIAR	-	MP RPA analysis results flag

/PROCON/

IFLT	-	Flight number
IDBEP	-	MP/EP results use flag 1: Use results from EPPRC 2: Use results from MPPRC (EP)
IDBRP	-	MP/RPA results use flag 1: Use results from RPAPRC 2: Use results from MPPRC (RPA)
ISWNE	-	Primary source of electron density 1: SM sensor 2: DM sensor 3: EP sensor (mode C or D only)
ISWVP	-	Primary source for vehicle potential
ISWCRL	-	Primary source of data for C_kL

/RPAANL/

IRPTME	-	Center time of RPA sweep (sec since midnight)
RPHDEN	-	RPA H+ ion density (ion/cm^3)
RPODEN	-	RPA O+ ion density (ion/cm^3)
RPITMP	-	RPA ion temperature (°K)
IRPLIF	-	Light ion flag
RPAUR	-	RPA ion ram velocity (meters/sec)
IRPAR	-	RPA analysis results flag

/SATEPH/

ITEDR	-	EDR times (sec since midnight)
IYMD	-	Date (YYMMDD)
GLAT	-	Geographic latitude (deg)
GLON	-	Geographic longitude (deg)
APXLAT	-	Apex latitude (deg)
APXLON	-	Apex longitude (deg)
APXLT	-	Apex local time (hours)
ALTSC	-	Satellite altitude (km)

/SCSOH/

ADCTMP	-	ADC electronics temperature
DMYO	-	DM sensor offset voltage reference (volts)
DSMTMP	-	DM/SM electronics temperature

/SMANL/

SMADEN	-	SM average ion density (ion/cm^3)
ISMAR	-	SM analysis results flag

c. Processing.

1. If the EDR load buffer (EDR) has just been initialized by Subroutine SETEDR (NPTEDR = -1), initialize all internal flags and jump down to step 6.

2. If this is a reset for a new REV, set the load buffer pointer (NPTEDR) to the other section of the buffer.

3. If the current data frame is bad, jump down to step 7.

4. If the current time (in IDTIME) is beyond the time span of the current EDR, then:

   (a) Determine the location of IDTIME with respect to the times in the ephemeris arrays,

   (b) Set the DONE/GONE control flags as follows:

      (1) If the EDR in the inactive section of the EDR load buffer has been output to the EDR I/O buffer (GONE = .TRUE.), flip the EDR load buffer pointer. If the current time is less than six seconds past the end time of the last EDR, set both DONE and GONE to .FALSE., otherwise set both to .TRUE..

(2) If the EDR in the inactive section has not been output (which indicates a large time jump in the current IDTIME), do not flip the pointer (thus overwriting the latest EDR which will contain much less data than the EDR in the inactive section) and set DONE and GONE to .TRUE..

5. If the current time is within the time covered by the current EDR, and the last EDR has not yet been output and the current time is more than six seconds beyond the end of the last EDR, set DONE and GONE to .TRUE.. Otherwise, set DONE to .FALSE..

6. If the new EDR is required (from step 4), load missing-data indicators into the EDR work array and load the ephemeris data, miscellaneous information, and engineering data into the new EDR. If the EDR is not new, check for a change in EP mode to/from a non-sweep (DC) mode. If the mode has changed, reinitialize the EP section of the current EDR for the new mode output.

7. If an analysis is available from the CKL module, load it into the appropriate EDR.

8. If the MP analyses for the EP or RPA sweeps are to be loaded in the EDR, and they are available, load the analyses as required (Subroutine LDSWPS).

9. If the current frame is bad, return to Subroutine OUTPUT.

10. Load the spacecraft potential.

11. Load the density data from the SM sensor.

12. Load the density and drift velocity data from the DM sensor.

13. If the EP analyses from module EPPRC are to be used in the EDRs, and an analysis is available, load the EP data (Subroutine LDSWPS).

14. If the RPA analyses from module RPAPRC are to be used in the EDRs, and an analysis is available, load the RPA data (Subroutine LDSWPS).

15. Return to Subroutine OUTPUT.

d. Outputs

1. Argument list.

DONE	-	Indicator that an EDR is ready for output
GONE	-	Indicator that an EDR has been output

2. COMMON blocks.

/EDRREC/

NPTEDR	-	EDR load buffer pointer
EDR	-	EDR load buffer

e. Associated subroutines.

1. Subroutine LDSWPS. Load an EP or RPA sweep analysis into the EDR load buffer (see

2.4.5.4.1).

f. Interfaces.

1. Called from Subroutine OUTPUT.

2. Uses COMMON blocks CKLANL, DFRAME, DMANL, EDRREC, EPANL, MPEANL, MPIANL, PROCON, RPAANL, SATEPH, SCSOH, and SMANL.

2.4.5.4.1 Subroutine LDSWPS.

a. Function.

Load data from either EP or RPA sweep analyses into an EDR.

b. Inputs.

1. Argument list.

ITYPE	-	Sweep type (1: EP, 2: RPA)
IESEC	-	Time of current EDR buffer (seconds since midnight)
ITIME	-	Sweep center time (seconds since midnight)
DEN1	-	Electron or O+ density
DEN2	-	Light ion density
TEMP	-	Electron or ion temperature
POT	-	Satellite potential
LIF	-	Light ion flag
UR	-	Ram ion drift velocity
IQUAL	-	Analysis qualifier
IEND	-	Location of EP sweeps (DC modes only)

2. COMMON blocks.

/EDRREC/

NPTEDR	-	EDR load buffer pointer

c. Processing.

1. Determine in which EDR in the EDR load buffer the current sweep should be stored, and the location within that EDR (N).

2. If an EP sweep, adjust N for DC mode sweeps and load the sweep analysis in the EDR.

3. If an RPA sweep, load the sweep analysis in the EDR.

4. Return to Subroutine LDEDR.

d. Outputs.

1. COMMON blocks.

/EDRREC/

EDR	-	EDR load buffer

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine LDEDR.

2. Uses COMMON block EDRREC.

2.4.6 Subroutine SUMOUT.

a. Function.

Build all end-of-run summary prints as necessary.

b. Inputs.

1. COMMON blocks.

/IONUM/

LUSTAT	-	Logical unit for file IESSTATFILE

/LUDIAG/

LUMPEP	-	Logical unit for MPPRC EP comparison output
LUMPRP	-	Logical unit for MPPRC RPA comparison output
LUDMHP	-	Logical unit for DM H+ mode output

/PROCON/

INTSUM	-	Processing statistics summary interval (hours)
IMPDGP	-	MPPRC diagnostic output flag
IRMIN	-	Wall time of current run (IES minutes)

2. Files.

IESSTATFILE. The REV average processing statistics are read in from file IESSTATFILE for display if the time since the last display is greater than the user set display interval.

IESDMHPMODE. The print from this diagnostic output file for DM H+ mode data is transferred to the HSP file if user-requested.

IESMPEPPRT. The print from this diagnostic output file, containing comparisons of MP analyses of EP sweeps to EP sweep analyses from module EPPRC, is transferred to the HSP file if user-requested.

IESMPRPAPRT. The print from this diagnostic output file, containing comparisons of MP analyses of RPA sweeps to RPA sweep analyses from module EPPRC, is transferred to the HSP file if user-requested.

c. Processing.

1. Check the last-time-displayed variable (LTMDIS) in all three sections of file IESSTATFILE. If the time since the last display is less than the interval between summaries specified by the user (variable INTSUM), generate a full summary for that section (Subroutine SFSUM).

2. If any data have been output to the DM H+ mode diagnostic output file (IESDMHPMODE), transfer the output to HSP (Subroutine IESPRT).

3. If the user has specified that the MP sweep analysis comparison files (IESMPEPPRT and IESMPRPAPRT) are to be printed in the HSP, transfer any records in these files to the HSP (Subroutine IESPRT).

4. Return to the main routine.

d. Outputs.

(See Subroutines SFSUM (2.4.6.1), MPDIAG (2.4.4.5.3), and HPMODE (2.4.4.4.7) for the format of the various HSP outputs generated.)

1. Files.

IESSTATFILE. If the statistics summary print is displayed, the updated display time variable (LTMDIS) is written out to each section of the file displayed.

e. Associated subroutines.

1. Subroutine SFSUM. Generate and output a summary of the records from file IESSTATFILE (see 2.4.6.1).

2. Subroutine IESPRT. Transfer ASCII-format print records from a diagnostic output file to HSP (see 2.4.6.2).

3. Subroutine FILERR. File error diagnostic control routine (see 2.4.8.5).

4. Subroutine PRNTON. Disable the PRINTAWAY processor and set the console error message flag (see 2.4.8.11).

f. Interfaces.

1. Called from the main routine.

2. Uses COMMON blocks IONUM, LUDIAG, and PROCON.

2.4.6.1 Subroutine SFSUM.

a. Function.

Generate and output an HSP display of the contents of an IESSTATFILE summary record.

b. Inputs.

1. Argument list.

HEADER	-	Entry for output header (CHARACTER)
MISSID	-	Satellite mission ID
IFLT	-	Satellite flight ID
ISUM	-	Summary statistics array (INTEGER)
RSUM	-	Summary statistics array (REAL, Equivalenced to ISUM)
NPRT	-	Number of readout summaries to print

c. Processing.

1. Print out the header and information in the first output block.

2. Print out the information in the second block.

3. Print out the information in the third block.

4. Return to the calling routine.

d. Outputs.

Figure 18 and Table 2 show an example of the HSP display generated for a single summary record, and Figure 19 shows an example of the output generated for a full file display.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutines SUMOUT and LDSTF.

2. Uses no COMMON blocks.

2.4.6.2 Subroutine IESPRT.

a. Function.

Read diagnostic output from an ASCII format file and transfer the records to the standard HSP file.

```
SSIES READOUT PROCESSING SUMMARY: SATELLITE F11  (MISSION ID: WX2546)

             +----------- PLASMA DENSITIES -----------+- TE -- TI -+-- RMS1 --+---  CKL ---+- DRIFT  VELOCITY -+
REV    DTG   I    EP   S M   RPA   S     DM       SM  I            I          I          S I   UH     UV   UR  I
-----+-------+------------------------------------------+------------+----------+------------+------------------+
  0 25/0959 I -999999 2 2   23483 1    46926    29101 I*****  2258 I 7.41E+09 I 5.02E+32 2 I  -253    -81  -371 I
-----+-------+------------------------------------------+------------+----------+------------+------------------+

   a    b        c    d e     f  g      h        i     j    k       l        m     n      o     p     q

             +---- POTENTIALS ---+--- SENSOR --+- DM -+------ PROCESSING STATS ------+----- ERROR FLAGS -----+- DUPE-+
             I                   I     TEMP    I VOLT I    RPA        EP        CKL      I                           I CYCLE I
REV    DTG   I  VIP  VBIAS S  VEP  I ELC ADC DSM I      I GOOD  ALL  GOOD  ALL  GOOD  ALL I RPA  EP  DM  SM  MP  CKL I COUNT I
-----+-------+-------------------+-------------+------+-----------------------------+--------------------------+-------+
  0 25/0959 I  0.00   15.5 0****** I  25  34 -48 I 2.53 I  917/1770     0/   0  327/ 702 I  0   0   0   0   0   1  I     1 I
-----+-------+-------------------+-------------+------+-----------------------------+--------------------------+-------+

                r   s   t   u     v   w   x   y        z          aa       bb       cc  dd  ee  ff  gg  hh       ii

             +------------ DENSITY DATA COMPARISONS -------------+
             +-------- SM TO -----------+---- DM TO ----+- EP TO -+-TE/TI-+
REV    DTG   I   DM      EP       RPA   I    EP      RPA  +   RPA   I  BAD  I
-----+-------+--------------------------+---------------+---------+-------+
  0 25/0959 I  -16759 -999999    1873 I -999999    4540 I -999999 I   0  I
-----+-------+--------------------------+---------------+---------+-------+

                jj     kk       ll      mm       nn       oo      pp
```

Figure 18 Processing statistics summary output sample - single readout

```
SSIES PERIODIC PROCESSING SUMMARY: SATELLITE F11  (MISSION ID: WX2546)

              +----------- PLASMA DENSITIES -----------+- TE -- TI -+-- RMS1 --+---  CKL ---+- DRIFT  VELOCITY -+
REV    DTG    I    EP   S M    RPA   S     DM       SM  I            I          I          S I  UH    UV    UR  I
-----+--------+-----------------------------------------+------------+----------+------------+-------------------+
   0 25/0959  I -999999 2 2   23483 1    46926    29101 I*****  2258 I 7.41E+09 I 5.02E+32 2 I  -253   -81  -371 I
-----+--------+-----------------------------------------+------------+----------+------------+-------------------+

              +---- POTENTIALS ---+--- SENSOR --+- DM -+------ PROCESSING  STATS -----+----- ERROR  FLAGS -----+- DUPE-+
              I                   I     TEMP    I VOLT I   RPA       EP        CKL    I                         I CYCLE I
REV    DTG    I  VIP  VBIAS S  VEP I ELC ADC DSM I      I GOOD  ALL  GOOD  ALL  GOOD  ALL I RPA  EP  DM  SM  MP  CKL I COUNT I
-----+--------+--------------------+-------------+------+-----------------------------+-------------------------+-------+
   0 25/0959  I  0.00  15.5 0****** I  25  34 -48 I 2.53 I  917/1770     0/   0   327/ 702 I  0   0   0   0   0    1 I     1 I
-----+--------+--------------------+-------------+------+-----------------------------+-------------------------+-------+

              +------------ DENSITY DATA COMPARISONS -------------+
              +--------- SM TO -----------+---- DM TO ----+- EP TO -+-TE/TI-+
REV    DTG    I   DM      EP       RPA   I    EP     RPA  +   RPA   I  BAD  I
-----+--------+--------------------------+---------------+---------+-------+
   0 25/0959  I  -16759 -999999    1873  I -999999   4540 I -999999 I   0   I
-----+--------+--------------------------+---------------+---------+-------+
```

Figure 19 Processing statistics summary output sample - periodic summary

Table 2 Summary print output description

(Labels cross-reference to Figure 18)

Label	Description
a	Readout REV number
b	Readout REV start date (DD) and time (HHMM) (GMT)
c	Average electron density from EP data analyses
d	Source of EP sweep analyses 1: SSIES program 2: On-board microprocessor
e	EP sensor operating mode (0-6 corresponds to modes A,B,BS,C,D,DS,E)
f	Average total ion density from RPA data analyses
g	Source of RPA sweep analyses 1: SSIES program 2: On-board microprocessor
h	Average ion density from DM LLA/LLB data
i	Average ion density from SM data
j	Average electron temperature from EP sweep analyses
k	Average ion temperature from RPA sweep analyses
l	Average power spectral density from first SM filter
m	Average C_kL
n	Source of data used in C_kL calculation 1: SM density data only 2: SM density and filter data 3: EP density data only
o	Average horizontal cross-track ion drift velocity (DM)
p	Average vertical cross-track ion drift velocity (DM)
q	Average ram ion drift velocity (RPA)
r	Average V_{IP}
s	Average V_{bias}
t	Source of V_{bias} 1: Set from MP analyses of EP sweep data 2: Set from SENPOT sensor
u	Average spacecraft potential from EP sweep analyses
v	Average electrometer instrument temperature
w	Average A/D converter instrument temperature
x	Average DSM instrument temperature
y	Average DM offset voltage
z	Number of successful RPA sweep analyses and number attempted
aa	Number of successful EP sweep analyses and number attempted
bb	Number of successful C_kL calculations and number attempted
cc-hh	Error/warning flags from processor modules RPA - none currently EP - none currently SM - DM - MP - none currently CKL -
ii	Number of duplicate data frames encountered
jj	Average difference between SM and DM ion densities
kk	Average difference between SM ion and EP electron densities
ll	Average difference between SM and RPA ion densities
mm	Average difference between DM ion and EP electron densities
nn	Average difference between DM and RPA ion densities
oo	Average difference between EP electron and RPA ion densities
pp	Number of cases when $T_e < T_i$

b. Inputs.

1. Argument list.

LU	-	Logical unit of diagnostic output file to read
NAME	-	Name of diagnostic output file (CHARACTER)

2. Files.

ASCII format records (132 characters per line maximum) are input from the diagnostic output file designated.

c. Processing.

1. Rewind the specified diagnostic output file.

2. If the first attempt to read from the file fails, write out a diagnostic statement and return to Subroutine SUMOUT.

3. Loop through the file, reading in an output line and printing it until all records in the diagnostic output file have been transferred.

4. Return to Subroutine SUMOUT.

d. Outputs.

The HSP output generated by this routine is in the same format as the data stored in the diagnostic output file read.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutine SUMOUT.

2. Uses no COMMON blocks.

2.4.7 Subroutine QUIT.

a. Function.

Specify which of the AGDB transfer files (always 1, see Subroutine OPNXFR 2.4.2.7 for comments) is to be transferred in the runstream, disable the PRINTAWAY processor in the APGA program runstream if required, send any necessary messages to the system operator's console, and close all files used by the APGA program.

b. Inputs.

1. COMMON blocks.

/IONUM/

LUEDR	-	Logical unit for file IESEDRFILE
LUXFR	-	Logical unit for file IESAGDBXFRn
LUSTAT	-	Logical unit for file IESSTATFILE

/LUDIAG/

LUSMA	-	Logical unit for SMPRC ASCII diagnostic output
LUSMB	-	Logical unit for SMPRC binary diagnostic output
LUMPEP	-	Logical unit for file IESMPEPPRT
LUMPRP	-	Logical unit for file IESMPRPAPRT
LUEPA	-	Logical unit for EPPRC ASCII diagnostic output
LUEPB	-	Logical unit for EPPRC binary diagnostic output
LURPAA	-	Logical unit for RPAPRC ASCII diagnostic output
LURPAB	-	Logical unit for RPAPRC binary diagnostic output
LUDMA	-	Logical unit for DMPRC ASCII diagnostic output
LUDMB	-	Logical unit for DMPRC binary diagnostic output
LUDMHP	-	Logical unit for DM H+ mode diagnostic output
LUFIBA	-	Logical unit for DM FIBA mode diagnostic output
LUCKLA	-	Logical unit for CKLPRC ASCII diagnostic output
LUCKLB	-	Logical unit for CKLPRC binary diagnostic output
LUQCA	-	Logical unit for QCPRC ASCII diagnostic output
LUEDRA	-	Logical unit for WRTEDR ASCII diagnostic output
LUXFRA	-	Logical unit for WRTXFR ASCII diagnostic output

/PROCON/

IAGDBP	-	Transfer file diagnostic output flag
ICNERR	-	Console message flag (see Figure 20)
ICNWRT	-	Console message write disable switch = 1: Do not send console messages ≠ 1: Send console messages
ICSDGP	-	CKLPRC module diagnostic print flag
IDMDGP	-	DMPRC module diagnostic print flag
IEDDGP	-	EDR diagnostic print flag
IEPDGP	-	EPPRC module diagnostic print flag
IMPOGP	-	MPPRC module diagnostic print flag
IPAWAY	-	PRINTAWAY disable flag = 0: Do not disable PRINTAWAY ≠ 0: Disable PRINTAWAY
IQCDGP	-	QCPRC module diagnostic print flag
IRPDGP	-	RPAPRC module diagnostic print flag
ISMDGP	-	SMPRC module diagnostic print flag
NFXFR	-	Transfer file number (always 1)

c. Processing.

1. Set the condition word (obsolete for SFC) and designate transfer file 1 for printing.

2. If the console message flag (ICNERR) is non-zero, and the console message disable flag (ICNWRT) is not one (1), write the messages shown in Figure 20, based on the value of ICNERR, to the output log.

3. Close all production files used.

4. Close any diagnostic output files used.

5. Return to the main routine.

d. Outputs.

The messages shown in Figure 20 are sent to HSP.

ICNERR	Message Generated
1	** ERROR WITH SYSTEM FILE ACFPARAMETER ** EXAMINE SSIES PRINT FOR ERROR **
2	** ERROR WITH SYSTEM FILE IESPREPFILE ** EXAMINE SSIES PRINT FOR ERROR **
3	** ERROR WITH SYSTEM FILE IESEDRFILE ** EXAMINE SSIES PRINT FOR ERROR **
4	** ERROR WITH SYSTEM FILE IESCNTRLFILE ** EXAMINE SSIES PRINT FOR ERROR **
5	** ERROR WITH SYSTEM FILE IESAPEXTABLE ** EXAMINE SSIES PRINT FOR ERROR **
6	** ERROR WITH SYSTEM FILE IESAGDBXFR1 ** EXAMINE SSIES PRINT FOR ERROR ** or ** ERROR WITH SYSTEM FILE IESAGDBXFR2 ** EXAMINE SSIES PRINT FOR ERROR **
7	** ERROR WITH SYSTEM FILE IESSTATFILE ** EXAMINE SSIES PRINT FOR ERROR **
8	** ERROR ENCOUNTERED IN SSIES PROCESSOR **
9	** EXAMINE SSIES PRINT FOR WARNING **

Figure 20 Console error messages generated from Subroutine QUIT

e. Associated subroutines.

None.

f. Interfaces.

1. Called from the main routine.

2. Uses COMMON blocks IONUM, LUDIAG, and PROCON.

2.4.8 General Purpose Library Routines.

2.4.8.1 Subroutine ADATE.

a. Function.

Retrieve the current system data and time.

b. Inputs.

None.

c. Processing.

1. Call the VAX/VMS intrinsic routine IDATE to get the current month, day and year.

2. Convert month, day, and year to packed CHARACTER representation (MMDDYY).

3. Call the VAX/VMS intrinsic routine TIME to get the current time.

4. Reformat the returned CHARACTER representation of current time to eliminate the colons (:) in the format.

d. Outputs.

1. Argument list.

MMDDYY -	Packed CHARACTER representation of Month (MM), Day (DD) and Year (YY)
HHMMSS -	Packed CHARACTER representation of Hours (HH), Minutes (MM), and Seconds (SS)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines.

2. Uses no COMMON blocks.

2.4.8.2 Function BITON.

a. Function.

Determine if a designated bit in the lowest order 9 bits is set in an input integer word.

b. Inputs.

1. Argument list.

IWORD	-	Word to check
IBIT	-	Bit number to check (1-9)

c. Processing.

1. See if the bit corresponding to 2**(IBIT-1) is set. If so, set BITON to .TRUE.; otherwise, set it to .FALSE.. (Note: The array TWON contains the values of 2**(IBIT-1) from IBIT=1 to IBIT=9.) (Note: The Boolean AND function was not used, as it is considered an extension to the ANSI Fortran-77 standard.)

2. Return to the calling routine.

d. Outputs.

1. Functional value.

BITON	= .TRUE. - If bit IBIT in word IWORD is set.
	= .FALSE. - If the bit is not set.

e. Associated subroutines.

None.

f. Interfaces.

1. Called from many routines.

2. Uses no COMMON blocks.

2.4.8.3 Subroutine COPY.

a. Function.

Copy information from source to destination.

b. Inputs.

1. Argument list.

SOURCE	-	Scalar or array containing the information to be copied

INCSRC	-	Increment used to step through the source array When INCSRC = 0, perform a scalar copy. When INCSRC > 0, beginning with the first value, copy every INCSRCth value into the DEST array.
INCDEST	-	Stepping increment for the DEST array as information is copied
NUMVALS	-	Number of values to copy from SOURCE to DEST

c. Processing.

1. When the source increment (INCSRC) is zero, propagate the scalar SOURCE into NUMVALS of the DEST array.

2. When the source increment (INCSRC) is non-zero, copy NUMVALS of the SOURCE array, beginning with the first value and incrementing by INCSRC, into the DEST array, at increments of INCDEST.

d. Outputs.

1. Argument list.

DEST	-	The destination array for copied information.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by many routines.

2. Uses no COMMON blocks.

2.4.8.4 Function ERF.

a. Function.

Calculate the value of the error function for the input argument. The algorithm used is from the National Bureau of Standards Handbook of Mathematical Functions, page 297, series expansion # 7.1.6.

b. Inputs.

1. Argument list.

X	-	Argument for the error function.

c. Processing.

1. If the argument is within 0.001 of zero, set ERF to the value of the input argument.

2. If the argument is outside the range [$-4 \leq X \leq +4$]:
set ERF to -1.0 when $X < -4$,
set ERF to +1.0 when $X > +4$.

3. Otherwise, calculate it from the series expansion referenced above.

4. Return to the calling routine.

d. Outputs.

1. Function value.

ERF	-	Value of the error function for X

e. Associated subroutines.

None.

f. Interfaces.

1. Called by several routines.

2. Uses no COMMON blocks.

2.4.8.5 Subroutine FILERR.

a. Function.

Fortran I/O error contingency routine.

b. Inputs.

1. Argument list.

FNAME	-	Name of file (CHARACTER)
IOTYPE	-	Type of I/O attempted 1: OPEN, 2: READ, 3: WRITE
IOERR	-	Error status returned from I/O attempt
NREC	-	Record number (direct access READ/WRITE only)
NFILE	-	Internal file number (1-7, for ICNERR and QUIT) 1: ACFPARAMETER 2: IESPREPFILE 3: IESEDRFILE 4: IESCNTRLFILE 5: IESAPEXTABLE 6: IESAGDBXFRn 7: IESSTATFILE NOTE: These file numbers do not correlate with the logical unit numbers assigned in Common IONUM!
RNAME	-	Name of subroutine calling FILERR (CHARACTER)

c. Processing.

1. Set the PRINTAWAY disable flag (IPAWAY) to one (1).

2. Unpack the I/O error status word (Sperry Fortran packs the status in the lower half of the 36-bit word).

   NOTE: This should be modified to reflect the VAX error reporting conventions.

3. If the attempt was to OPEN a file, write out the message:

   ERROR(rname) COULD NOT OPEN FILE fname
   STATUS: ioerr
   ***REFER TO OPERATOR REFERENCE GUIDE FOR REQUIRED ACTIONS**

   where:
   rname is the name of the routine in which the error occurred,
   fname is the name of the file to be OPENed,
   ioerr is the lower half of the error status.

   Set ICNERR to the internal file number, and return to the calling routine.

4. If the attempt was to WRITE to a file, write out the message:

   ERROR(rname) COULD NOT WRITE TO FILE fname
   STATUS: ioerr RECORD: nrec
   ***REFER TO OPERATOR REFERENCE GUIDE FOR REQUIRED ACTIONS**

   where:
   rname is the name of the routine in which the error occurred,
   fname is the name of the file to be written to,
   ioerr is the lower half of the error status,
   nrec is the I/O record to which the write was directed.

   (Note: If this was not a direct access write, the RECORD item is omitted.)

   Set ICNERR to the internal file number, and return to the calling routine.

5. If the attempt was to READ from a file, write out the message:

   ERROR(rname) COULD NOT READ FROM FILE fname
   STATUS: ioerr RECORD: nrec
   ***REFER TO OPERATOR REFERENCE GUIDE FOR REQUIRED ACTIONS**

   where:
   rname is the name of the routine in which the error occurred,
   fname is the name of the file to be read from,
   ioerr is the lower half of the error status,
   nrec is the I/O record from which the read was directed.

   (Note: If this was not a direct access read, the RECORD item is omitted.)

Set ICNERR to the internal file number, and return to the calling routine.

d. Outputs.

The above messages are written to HSP.

1. COMMON blocks.

/PROCON/

IPAWAY	-	PRINTAWAY disable flag
ICNERR	-	Console error flag

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines.

2. Uses COMMON block PROCON.

2.4.8.6 Subroutine FITLIN.

a. Function.

Perform a linear least-squares fit to an input set of x and y values.

b. Inputs.

1. Argument list.

X	-	x values
Y	-	y values
N	-	number of x-y pairs

c. Processing.

1. Calculate the elements of the least-squares matrix (the sums of x, y, x^2, and xy).

2. Calculate the determinant of the matrix from:

$$C = N \sum x^2 - (\sum x)^2$$

3. If the determinant is zero, set the linear fit coefficients to zero.

4. If not, calculate the fit coefficients from:

$$A = (N \sum xy - \sum x \sum y)/C$$

$$B = (\sum x^2 - \sum x \sum xy)/C$$

5. Return to the calling routine.

d. Outputs.

1. Argument list.

A	-	Slope of linear fit
B	-	Zero-intercept of linear fit

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines (RPAPRC module).

2. Uses no COMMON blocks.

2.4.8.7 Subroutine LATLON.

a. Function.

Calculate the geographic latitude and longitude of a satellite in a circular orbit for the input time.

b. Inputs.

1. Argument list.

IDT	-	Time since the time in ITCKL(1)

2. COMMON blocks.

/SATEPH/

PHI0	-	Angle in orbit at time ITCKL(1) (rad)
OMEGA	-	Satellite angular velocity (rad/sec)
ORBINC	-	Orbital inclination (rad)
ANLON	-	Longitude of last ascending node (rad)

c. Processing.

1. Calculate the angle in the orbit from the last ascending node for time IDT after ITCKL(1) (DPHI), and the total time since the last ascending node (DT).

2. Calculate the latitude of the satellite from:

$$\lambda = \sin^{-1} [\sin i \sin\Delta\phi]$$

where: λ is the latitude,
i is the orbital inclination (ORBINC),
$\Delta\phi$ is the angle in the orbit from ANLON.

3. Calculate the longitude of the satellite from:

$$\Theta = \tan^{-1} \frac{\cos i \sin \Delta\phi}{\cos \Delta\phi} + \Theta_a - \Omega_e \Delta t$$

where: Θ is the longitude,
Ω_e is the angular rotational velocity of the earth.

4. Return to the calling routine.

d. Outputs.

1. Argument list.

SLAT	-	Satellite latitude (deg)
SLON	-	Satellite longitude (deg)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from routines EPHEM and MPDIAG.

2. Uses COMMON blocks PIDEG and SATEPH.

2.4.8.8 Subroutine LOGLIN.

a. Function.

Convert a list of numbers to/from base-10 logarithms.

b. Inputs.

1. Argument list.

XIN	-	List of number to convert
N	-	Number of values in XIN

ITYPE	-	Direction of conversion 1: from log 2: to log

c. Processing.

1. If ITYPE is 1, convert the values in XIN from logs, setting all output values to zero for XIN ≥ 0.

2. If ITYPE is 2, convert the values in XIN to logs, setting all output values to zero for XIN ≤ 0.

3. Return to the calling routine.

d. Outputs.

1. Argument list.

XOUT	-	Array of XIN converted values

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines.

2. Uses no COMMON blocks.

2.4.8.9 Subroutine OPNDOF

a. Function.

Open diagnostic output files.

b. Inputs.

1. Argument list.

IDIAG	-	Diagnostic print flag 0: Write ASCII output to HSP file (unit 6) 1: Open binary output file 2: Open ASCII output file 3: Open both binary and ASCII output files
LUASC	-	Logical unit for ASCII output file
FNASC	-	Name of ASCII output file (CHARACTER)
LUBIN	-	Logical unit for binary output file
FNBIN	-	Name of binary output file (CHARACTER)
NRSIZE	-	Size of binary file output records (words)

c. Processing.

1. If the diagnostic print flag is negative, set it positive and set the ASCII output file unit to 6.

2. Determine which files to open based on the bits set in the diagnostic print flag.

3. Open the binary output file if requested. If an error is encountered, alert the user, and turn off the binary diagnostic output.

4. Open the ASCII output file if requested. If an error is encountered, alert the user, and turn off the ASCII diagnostic print.

5. Return to the calling routine.

d. Outputs.

1. Argument list.

ASCII	-	.TRUE. if ASCII output file OPENed
BINARY	-	.TRUE. if binary output file OPENed

e. Associated subroutines.

1. Function BITON. Determine if a bit is set in an integer word (see 2.4.8.2).

2. Subroutine FILERR. File error diagnostic control routines (see 2.4.8.5).

f. Interfaces.

1. Called from several routines.

2. Uses no COMMON blocks.

2.4.8.10 Function PRMRNG.

a. Function.

Check an environment parameter to see if it is within allowed range (ranges set in the BLOCK DATA routine, see 2.4.1.2).

b. Inputs.

1. Argument list.

PRM	-	Parameter to check
NPRM	-	Parameter number 1: Plasma density 2: Electron temperature 3: Ion temperature 4: Horizontal ion drift velocity 5: Vertical ion drift velocity

6: p_1
7: T_1
8: C_kL
9: Satellite potential

2. COMMON blocks.

/EDRNGS/

PARMAX	-	Maximum allowed values
PARMIN	-	Minimum allowed values
IPRM	-	Missing-data replacement flag: = 0: replace out-of-range values by -1.0E37 = 1: replace out-of-range values by nearest allowed extrema > 1: do not replace out-of-range values

c. Processing.

1. If the parameter is within range, set the function value to .TRUE..

2. If not, set the function value to .FALSE. and set the parameter value to -1.0E37 if the replacement flag is zero, or to the maximum or minimum value if the flag is one.

3. Return to the calling routine.

d. Outputs.

1. Function value.

PRMRNG	-	.TRUE. if the parameter is in range .FALSE. if the parameter is out of range

2. Argument list.

PRM	-	Set to -1.0E37 if out-of-range and the flag is 0 Set to max or min value if out of range and the flag is 1

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines.

2. Uses COMMON block EDRNGS.

2.4.8.11 Subroutine PRNTON.

a. Function.

Disable the PRINTAWAY flag and set variable ICNERR for use in Subroutine QUIT to generate error messages.

b. Inputs.

1. Argument list.

ICON	-	Value assigned to ICNERR

2. COMMON blocks.

/PROCON/

ICNERR	-	Console error message flag

c. Processing.

1. Set variable IPAWAY to one (1) to have Subroutine QUIT disable the PRINTAWAY processor.

2. If this is the first call to this routine, set ICNERR to ICON.

3. If not, set ICNERR to ICON only if ICON is less than ICNERR (indicating a possibly worse problem).

4. Return to the calling routine.

d. Outputs.

1. COMMON blocks.

/PROCON/

IPAWAY	-	PRINTAWAY disable flag
ICNERR	-	Console error message flag

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several subroutines.

2. Uses COMMON block PROCON.

2.4.8.12 Subroutine TIMCON.

a. Function.

Convert between date/time (GMT) and IES minutes, defined as the number of minutes since 0000GMT on 1 January 1985. This time measure is used by the APGA program for file control. Leap years, end-of-years, end-of-days, etc., are all handled by using IES minutes (similar in concept to Julian hours used by the AFGWC database system). Note that the choice of 1 January 1985 was totally arbitrary. This routine is good until the time standards are reset to pick up missing minutes.

b. Inputs.

1. Argument list.

IDIR	-	Conversion direction 1: to IES minutes 2: from IES minutes
IDATE	-	Date (YYMMDD) (input if IDIR = 1)
ITIME	-	Time (HHMM) (input if IDIR = 1)
IESMIN	-	IES minute (input if IDIR = 2)

c. Processing.

1. If converting from date/time to IES minutes, then:

(a) Determine the number of days since 1 January 1985
(b) Convert the days to minutes,
(c) Add on the number of minutes in the current day.

2. If converting from IES minutes to date/time, then:

(a) Determine the time in the current day,
(b) Determine the year,
(c) Determine the month and day,
(d) Calculate the date (YYMMDD).

3. Return to the calling routine.

d. Outputs.

1. Argument list.

IDATE	-	Date (YYMMDD) (output if IDIR = 2)
ITIME	-	Time (HHMM) (output if IDIR = 2)
IESMIN	-	IES minute (output if IDIR = 1)

e. Associated subroutines.

None.

f. Interfaces.

1. Called from several routines.

2. Uses no COMMON blocks.

2.4.8.13 Subroutine UPBITS.

a. Function.

Unpack the lower order bits from a word and load them into an array consisting of 1 bit per word.

b. Inputs.

1. Argument list.

IWORD	-	Word to unpack
NBITS	-	Number of bits to unpack

c. Processing.

1. Extract the lowest bit from the word and load it into the array.

2. Divide the word by two to drop off the lowest bit.

3. Repeat steps 1 and 2 until all bits have been extracted.

4. Return to the calling routine.

d. Outputs.

1. Argument list.

IARRAY	-	Array of unpacked bits

e. Associated subroutines.

None.

f. Interfaces.

1. Called from Subroutines MPEP and MPRPA.

2. Uses no COMMON blocks.

2.5 Program LDCON02.

a. Function.

Load the IESCNTRLFILE with the instrument and control parameters required to properly run

the APGA program. This program was developed for SSIES2 based upon a functional emulation of the LDCON program formerly used at AFGWC.

The program accommodates the entry of a full set of values for up to 3 different satellites. The structure of the control file and therefore the data dimensions and ordering in LDCON02 are closely allied to the size and structure of the APGA COMMON blocks SCNMOD, SATPAR, PROCON, SMCON, EPCON, RPACON, DMCON, CKLCON, MPCON, and DFCON. Any change to one of these COMMON blocks in APGA must be matched with an equivalent change in LDCON02 to keep the data in proper alignment. (See 2.4.2.4, Subroutine VEHPAR).

b. Inputs.

1. Argument list.

None.

2. COMMON blocks.

None.

3. Files.

IESCONLIST.ASCII - An optional input file containing the values to be written to the binary output file.

See 8.4 of Volume 2, Programmer's Guide, for a complete list of the LDCON02 parameter names, their APGA equivalent parameter names, and their default values for the IES2 F11 satellite.

c. Processing.

1. Inquire about the existence of the optional ASCII input file.

2. Open the binary output file.

3. If the optional ASCII input file does not exist, begin an interactive dialogue with the user to acquire the values for the control file. When the user enters the slash character (/) or **<CTRL-Z>**, the default value assigned in the internal data statement will be used for the specified variable.

4. If the optional ASCII file does exist, open the file and read the control file values from the file.

5. Write the input values to the binary output file for use by the APGA program.

6. If the optional ASCII input file did not exist, create one and write the user supplied values to the ASCII file.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

   None.

3. Files.

   IESCTL.DAT - A single record binary file containing the processing control information and instrument processing constants in the format required by the APGA program.

   IESCONLIST.ASCII - A formatted text file containing the values written to IESCTL.DAT.

e. Associated subroutines.

   1. Subroutine IGET. Interact with the user terminal to prompt for and read an Integer input value.

   2. Subroutine RGET. Interact with the user terminal to prompt for and read a Real input value.

f. Interfaces.

   1. Program initiated by user.

   2. Uses no COMMON blocks.

2.6 Program BNBA Description.

2.6.1 Main Routine.

a. Function.

   Control the unpacking, reordering, and reformatting of raw IES, IES2, IES2A, and IES3 data streams to produce a Common Format Prepfile for the APGA processing program.

b. Inputs.

   1. Files:

      TMFILE - File containing the raw packed IES, IES2, IES2A, or IES3 telemetry and ephemeris data.

c. Processing.

   1. Invoke Subroutine USERIN to obtain run control parameters from the user.

   2. Initialize data storage arrays.

   3. Invoke Subroutine RIREXT to open the input file and process the header record.

   4. Verify that the input file data type matches the requested data type.

BNBA:	Main Program
. USERIN:	Request run parameter inputs
.. CHEK_IN:	Verify run parameter inputs
. RIREXT:	Read, unpack, and reformat RIR data
.. ERRLOG:	I/O Error reporting
. EPHEXT:	Read and unpack Ephemeris data
.. ERRLOG:	I/O Error reporting
. TMEXT:	Read and unpack Telemetry data
.. ERRLOG:	I/O Error reporting
. WORKOUT:	Write unpacked Ephemeris/Telemetry data to intermediate work file
.. ERRLOG:	I/O Error reporting
. WORKIN:	Read unpacked Ephemeris/Telemetry data in time ascending order
.. ERRLOG:	I/O Error reporting
. PRINTIT:	List unpacked data to diagnostic file
. GET_XCEPTS:	Reformat Telemetry data
.. IES_XCEPT:	Process exceptions for IES data
... VALCHEK:	Verify Telemetry data
... CYCNT1:	Cycle 1 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT2:	Cycle 2 specific processing
... CYCNT2:	Cycle 2 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT1:	Cycle 1 specific processing
.. IES2_XCEPT:	Process exceptions for IES2 data
... VALCHEK:	Verify Telemetry data
... CYCNT1:	Cycle 1 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT2:	Cycle 2 specific processing
... CYCNT2:	Cycle 2 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT1:	Cycle 1 specific processing
.. IES2A_XCEPT:	Process exceptions for IES2A data
... VALCHEK:	Verify Telemetry data
... CYCNT1:	Cycle 1 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT2:	Cycle 2 specific processing
... CYCNT2:	Cycle 2 specific processing
.... VALCHEK:	Verify Telemetry data
.... CYCNT1:	Cycle 1 specific processing
.. IES3_XCEPT:	Process exceptions for IES3 data
. STOREM:	Store reformatted Telemetry block
. OUTPUT:	Write reformatted RIR/Ephemeris/Telemetry data blocks

Figure 21 Program BNBA hierarchy chart

5. Establish the unpacking loop to:

   a. read and unpack an Ephemeris data record,

   b. read and unpack a telemetry data record,

c. write the unpacked data to an intermediate work file,

d. loop through step 5 until all requested data have been unpacked.

6. If the unpacking was not successful, print an informative message and terminate the run. Otherwise:

   a. invoke Subroutine WORKIN to read the first (lowest time) unpacked data record,

   b. invoke Subroutine GET_XCEPTS to process the Common Format storage exceptions for the first data record.

7. Establish the loop to successively read and reformat each unpacked data record and write the results to the output file. Because the data reformatting involves the transfer of data from one cycle (record) to another, the input data is double buffered.

d. Outputs.

1. Files.

IESPREPFIL - The unpacked, reordered, and reformatted Ephemeris and telemetry data. This file is used as the primary input file to APGA.

WORKFIL - The keyed, indexed intermediate file for temporary storage of unpacked data. This allows the data records to be read in time ascending order for reformatting.

e. Associated subroutines.

1. Subroutine USERIN. Acquire user specified processing control parameters.

2. Subroutine RIREXT. Open the input file and read and unpack the header record.

3. Subroutine EPHEXT. Read and unpack an Ephemeris data record.

4. Subroutine TMEXT. Read and unpack a telemetry data record.

5. Subroutine WORKOUT. Write an unpacked Ephemeris and telemetry records to the intermediate file.

6. Subroutine WORKIN. Read an unpacked Ephemeris and telemetry record in time ascending order.

7. Subroutine GET_XCEPTS. Process the Common Format storage exceptions.

8. Subroutine STOREM. Store the telemetry data in the Common Format.

9. Subroutine OUTPUT. Write a reformatted data record to the output file.

f. Interfaces.

1. Main routine.

2. Uses COMMON blocks OUTREC, OUTRIR, and WORKREC.

2.6.2 Subroutine USERIN.

a. Function.

Prompt the user for necessary control parameters, and verify the validity of the values entered.

b. Inputs.

All inputs are parameters supplied by the user.

c. Processing.

1. Prompt the user to supply the input file path and name.

2. Prompt the user to supply the output file path and name.

3. Prompt the user to supply the work file path and name.

4. Prompt the user to supply the instrument type, and validity check the response (IES, IES2, IES2A, IES3).

5. Prompt the user to supply a 4 digit mission ID.

6. Prompt the user to supply a start time (Year, Day of year, Hour, Minute, Second) for the processing, and validity check the response.

7. Prompt the user to supply an end time (Year, Day of year, Hour, Minute, Second) for the processing, and validity check the response.

8. Query the user regarding the generation of a diagnostic data listing.

9. Echo the user supplied parameters and prompt the user for a response to proceed or modify the parameters.

10. Return to program BNBA.

d. Outputs.

1. Argument list.

TMFILENAM -	Full path name of the input file
IESPREPFIL -	Full path name of the output file
WORKFILNAM -	Full path name of the work file
IESTYP -	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3

MIDREQ	-	Specified mission ID
INTIME	-	Time to start processing (seconds)
NDTIME	-	Time to end processing (seconds)
PROCEED	-	Proceed with processing flag (LOGICAL)
LISTDATA	-	Diagnostic listing generation flag (LOGICAL)

e. Associated subroutines.

1. Subroutine CHEK_IN. Check validity of user supplied times, and convert Hour, Minute, Second to total seconds.

f. Interfaces.

1. Called by the main routine.

2. Uses no COMMON blocks.

2.6.3 Subroutine RIREXT.

a. Function.

Unpack, convert, and store the Readout Information Record (RIR) for use by the APGA program.

b. Inputs.

1. Argument list.

INUNIT	-	Input file unit number
TMFILENAM	-	Input file name
IESREQ	-	Instrument indicator code

2. COMMON blocks.

None.

3. Files.

TMFILE - The raw data file from which the header (RIR) record is read.

c. Processing.

1. Open the designated (TMFILENAM) input data file.

   a. If the open was not successful, set the proper indicators,

   b. Invoke ERRLOG to write an informative message,

   c. Return to BNBA.

2. Read the Readout Information Record into a VAX BYTE array.

a. If the read was not successful, set the proper indicators,

b. Invoke ERRLOG to write an informative message,

c. Return to BNBA.

3. Align the bit sequence properly for unpacking (i.e.: into the order originally written on the Sperry UNIVAC) by reversing the order of the VAX longwords in the data array which is EQUIVALENCEd to the input BYTE array.

4. Initialize the word (LWORD) and start bit within the word (INBIT) pointers to begin the unpacking process at the end of the bit stream.

5. Establish a loop to unpack each of the 40 data quantities needed, using the VAX intrinsic function IBITS. The size of each quantity (in bits) is specified in the array NBITS, the number of bits to skip between quantities is specified in the array NSKIP, and the word and start bit within the word pointers are decremented as needed.

6. Store the 4 characters of the Mission ID into a VAX longword using the VAX intrinsic function MVBITS.

7. Concatenate the 12 characters of the prepfile name into a VAX character array using the VAX intrinsic function MVBITS.

8. Determine the sensor type (IES, IES2, IES2A, or IES3) from the prepfile name, and store the proper code into the output array.

9. Convert the Mission ID to an integer and store it into the output array.

10. Calculate and properly format the remaining data needed by APGA and store it into the output array.

11. Return to BNBA.

d. Outputs.

1. Argument list.

IRCDKEY -	Index key value from RIR record read
IESTYP -	Instrument code:
	1 =IES, 2 = IES2, 3 = IES2A, 4 = IES3
MISSID -	Mission ID (integer)
IRIRRTN -	I/O status flag:
	= 0: No error
	> 0: Open or Read error occurred

2. COMMON blocks.

/OUTRIR/

IRIROUT - Array of unpacked and reformatted RIR data

e. Associated subroutines.

1. Subroutine ERRLOG. I/O error reporting.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block OUTRIR.

2.6.4 Subroutine EPHEXT.

a. Function.

Unpack, convert, and store the Ephemeris data for use by the APGA processing program.

b. Inputs.

1. Argument list.

INUNIT	-	Input file unit number
IRCDKEY	-	Index key value from last input record read
IESTYP	-	Instrument indicator code: 1 =IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current data record
IPREV	-	Pointer to previous data record

3. Files.

TMFILE - The raw data file from which the Ephemeris data record is read.

c. Processing.

1. Read the Ephemeris data record into a VAX BYTE array.

   a. If the read was not successful set the proper indicators,

   b. Invoke ERRLOG to write an informative message,

   c. Return to BNBA.

2. Align the bit sequence properly for unpacking (i.e.: into the order originally written on the Sperry UNIVAC) by reversing the order of the VAX longword array which is EQUIVALENCEd to the input BYTE array.

3. Initialize the word (LWORD) and start bit within the word (INBIT) pointers to begin the

unpacking process at the end of the bit stream.

4. Establish a loop to unpack each of the 28 data quantities needed, using the VAX intrinsic function IBITS. The size of each quantity (in bits) is specified in the array NBITS, the number of bits to skip between quantities is specified in the array NSKIP, and the word and start bit within the word pointers are decremented as needed.

   a. When the quantity being unpacked is in a Sperry floating point (REAL) form, the sign, characteristic, and mantissa are unpacked as separate entities. The characteristic and mantissa are one's complemented when the sign is negative, and the characteristic is normalized for the Sperry exponent convention and the number of bits in the mantissa (27). The VAX equivalent floating point value is calculated from the components.

   b. When the quantity being unpacked is INTEGER, the least significant 32 bits (of the Sperry 36 bit integer) are extracted.

   c. The converted value is stored into the current output array.

5. Return to BNBA.

d. Outputs.

1. Argument list.

IRCDKEY	-	Index key value from Ephemeris record read
IEPHRTN	-	Read status: = 0: No error > 0: Read error occurred

2. Common blocks.

/WORKREC/

IPHVAL	-	Array of unpacked and reformatted Ephemeris data

e. Associated subroutines.

1. Subroutine ERRLOG. I/O error reporting.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

2.6.5 Subroutine TMEXT.

a. Function.

Unpack, convert, and store the 60 sets of 1 second Telemetry data for use by the APGA processing program.

b. Inputs.

1. Argument list.

INUNIT	-	Input file unit number
IRCDKEY	-	Index key value from last input record read
IESTYP	-	Instrument indicator code: 1 =IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current data record
IPREV	-	Pointer to previous data record

3. Files.

TMFILE - The raw data file from which the Telemetry data record is read.

c. Processing.

1. Read the Telemetry data record into a VAX BYTE array.

   a. If the read was not successful, set the proper indicators,

   b. Invoke ERRLOG to write an informative message,

   c. Return to BNBA.

2. Align the bit sequence properly for unpacking (i.e.: into the order originally written on the Sperry UNIVAC) by reversing the order of the VAX longword array which is EQUIVALENCEd to the input BYTE array.

3. Initialize the word (LWORD) and start bit within the word (INBIT) pointers to begin the unpacking process at the end of the bit stream.

4. Establish a loop to unpack each of the 60 sets of Telemetry data quantities needed, using the VAX intrinsic function IBITS. The size of each quantity (in bits) is specified by NBITS, and the pointers to the word and to the start bit within the word are decremented as needed.

   a. Establish a loop to unpack the (instrument dependent) number of sets of data each second. Note that a set is defined as 4, 9 bit quantities, which equals 1 Sperry 36 bit word.

   b. The first set of data for each second contains the coded time for the record. These values are extracted and concatenated into a VAX longword.

   c. The remaining sets each contain 4, 9 bit coded instrument data, or control information. Each of these 9 bit values is extracted, aligned, and stored into the current output (VAX INTEGER * 2) array.

   d. The time value is converted into seconds and stored into the current time array.

5. When the data for all 60 seconds have been processed, return to BNBA.

d. Outputs.

1. Argument list.

IRCDKEY	-	Index key value from Telemetry record read
ITMRTN	-	Read status: = 0: No error > 0: Read error occurred

2. COMMON blocks.

/WORKREC/

ITMSEC	-	Array of 1 second telemetry times
ITMUPK	-	Array of unpacked telemetry data

e. Associated subroutines.

1. Subroutine ERRLOG. I/O error reporting.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

2.6.6 Subroutine WORKOUT.

a. Function.

Write the unpacked Ephemeris and Telemetry data to an intermediate storage file.

b. Inputs.

1. Argument list.

WORKFILNAM	-	Full path name for the work file
IWRKUNIT	-	Unit number of the work file
IESTYP	-	Instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current data buffer
IPHVAL	-	Unpacked Ephemeris data
ITMSEC	-	Telemetry data times
ITMUPK	-	Unpacked Telemetry data

c. Processing.

1. If this is the first time through:

   a. Set the record length (LRECL) and the size of the Telemetry data block (NTMWDS), based upon the instrument type, and initialize the primary indexed write key.

   b. Open the keyed, indexed file, defining the primary key as the sequential record count and the alternate key as the location of the Ephemeris time in each record.

   c. If the open was not successful, set the proper indicators, invoke ERRLOG to write an informative message, and return to BNBA.

2. Increment the primary (output) key.

3. Write the current set of Ephemeris, time, and telemetry data to the work file.

4. If the write was not successful, set the proper indicators, and invoke ERRLOG to write an informative message.

5. Return to BNBA.

d. Outputs.

1. Argument list.

KEYOUT -	Number of last record written to file
IWOSTAT -	I/O status value:
	= 0: No error
	> 0: Open or write error occurred

2. Files.

WORKFIL - The temporary file to which the current 1 minute set of data is written.

e. Associated subroutines.

1. Subroutine ERRLOG. I/O error reporting.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

2.6.7 Subroutine WORKIN.

a. Function.

Read the unpacked Ephemeris/Telemetry data records from the intermediate storage file in ascending time order by using the alternate key.

b. Inputs.

1. Argument list.

WORKFILNAM	-	Full path name for the work file
IWRKUNIT	-	Unit number of the work file
IESTYP	-	Instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current data buffer

3. Files.

WORKFIL - The temporary file from which the current 1 minute set of data is read.

c. Processing.

1. If this is the first time through:

   a. Set the record length (LRECL) and the size of the Telemetry data block (NTMWDS) based upon the instrument type.

   b. Open the keyed, indexed file with the primary key as the sequential record count and the alternate key pointing to the Ephemeris time in each record.

   c. If the open was not successful, set the proper indicators, invoke ERRLOG to write an informative message, and return to BNBA.

   d. Establish the use of the alternate key and read the earliest (in time) set of Ephemeris, time, and Telemetry data into the current (ICUR) buffers.

2. Otherwise, read the next (in ascending time order) data record.

3. If the read was not successful, set the proper indicators, and invoke ERRLOG to write an informative message.

4. Return to BNBA.

d. Outputs.

1. Argument list.

IWISTAT	-	I/O status value: = 0: No error > 0: Open or read error occurred

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current minute data buffer
IPHVAL	-	Unpacked Ephemeris data
ITMSEC	-	Telemetry data times
ITMUPK	-	Unpacked Telemetry data

e. Associated subroutines.

1. Subroutine ERRLOG. I/O error reporting.

f. Interfaces.

1. Called by the main routine.

2. Uses COMMON block WORKREC.

2.6.8 Subroutine GET_XCEPTS.

a. Function.

Control the processing of telemetry data exceptions for all instrument types. An exception is defined as:

1. A data quantity whose Common Format data cycle differs from its instrument telemetry data cycle.

2. A quantity whose Common Format data rate differs from its instrument telemetry data rate.

3. A data quantity which is created by BNBA from instrument telemetry data and stored in the Common Format.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1 minute data record
IPREV	-	Pointer to previous 1 minute data record
IXVAL	-	Array of telemetry storage exception values

3. Files.

None.

c. Processing.

1. Initialize the exceptions array for the current minute of data to null values (-1).

2. Propagate any forward-filled data exceptions by copying the extra 4 cycles (61 - 64) from the previous minute's exception set into the first 4 cycles of the current minute's exception set.

3. Invoke the subroutine which processes exceptions for the specified instrument's telemetry data.

   a. When IESTYP is 1, invoke IES_XCEPT.
   b. When IESTYP is 2, invoke IES2_XCEPT.
   c. When IESTYP is 3, invoke IES2A_XCEPT.
   d. When IESTYP is 4, invoke IES3_XCEPT.

4. Return to BNBA.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/WORKREC/

IXVAL - Array of telemetry storage exception values

e. Associated subroutines.

1. Subroutine IES_XCEPT. Process exceptions for the IES instrument telemetry data.

2. Subroutine IES2_XCEPT. Process exceptions for the IES2 instrument telemetry data.

3. Subroutine IES2A_XCEPT. Process exceptions for the IES2A instrument telemetry data.

4. Subroutine IES3_XCEPT. Process exceptions for the IES3 instrument telemetry data.

f. Interfaces.

1. Called by BNBA.

2. Uses COMMON block WORKREC.

2.6.8.1 Subroutine IES_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES instrument configuration. The specific IES exceptions are listed in Table 3.

Table 3 IES exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	14	2	111/all	Ion Velocity
2	1	145	1	1/9	Cycle1 MSB
3	1	146	1	1/6-8	EP Mode
4	1	147	always=2 for IES		EP/RPA Flag
5	1	148	1	1/5	Test Mode
6	1	149	1	1/4	Bias Mode
7	1	150	1	1/3	Sweep Clock
8	1	151	1	1/2	PRF/Reset
9	1	155	1	90/2-3	VIP Setting
10	1	156	1	90/1-5	VBIAS Volts
11	1	158	1&2	1/	Cycle Count
1	2	15	1	61/all	DM Signal Level
2	2	145	1	1/9	Cycle1 MSB
3	2	146	1	1/6-8	EP Mode
4	1	147	always=2 for IES		EP/RPA Flag
5	2	151	2	70/9	VBias Monitor
6	2	155	2	90/7-8	VIP Setting
7	2	156	2	90/1-5	VBIAS Volts
8	2	158	1&2		Cycle Count

b. Inputs.

1. Argument list.

   None.

2. COMMON blocks.

   /WORKREC/

ICUR	-	Pointer to current 1 minute data buffers
IPREV	-	Pointer to previous 1 minute data buffers
ITMUPK	-	Unpacked telemetry data buffers

3. Files.

   None.

c. Processing.

1. Establish a loop to process each of the 60 one-second telemetry data blocks in the current 1 minute buffer.

2. Invoke Subroutine VALCHEK to validity check the specified second's worth of data.

   a. If the data are not valid, set the cycle indicator to the null value (-1), set the break indicator to .TRUE. to indicate a discontinuity, and proceed to the next second's worth of data.

   b. If the data are valid, proceed to step 3.

3. Calculate the data cycle from the Cycle ID and store it in the data cycle indicator.

4. If this is a Cycle 1 data set:

   a. Invoke Subroutine CYCNT1 to extract the exceptions from the Cycle ID and the Configuration ID words.

   b. Save the exceptions which appear in the Cycle 1 instrument telemetry data set.

5. If this is a Cycle 2 data set:

   a. Invoke Subroutine CYCNT2 to extract the exceptions from the Cycle ID and Configuration ID words.

   b. Save the exceptions which appear in the Cycle 2 instrument telemetry data set.

6. When all 60 cycles (seconds) have been processed, return to Subroutine GET_XCEPTS.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

   /WORKREC/

NCYCLE	-	Array of Cycle indicators
IXVAL	-	Array of saved exceptions

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT1. Process Cycle ID and Configuration ID for Cycle 1.

3. Subroutine CYCNT2. Process Cycle ID and Configuration ID for Cycle 2.

f. Interfaces.

1. Called by Subroutine GET_XCEPTS.

2. Uses COMMON block WORKREC.

2.6.8.1.1 Subroutine VALCHEK.

a. Function.

Check the validity of the unpacked telemetry data for the specified second.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1 minute data record
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. Initialize the cycle indicator for this second of the current minute's telemetry data buffer.

2. When the telemetry Cycle ID is a non-zero positive number, the data are valid.

3. When the Cycle ID is zero or negative, the data are not valid when the following are all zero or negative:

a. For the IES instrument:

EP Sweep Voltage
RPA Sweep Voltage
SM (WIBAN1) Range
DM Log Level Amplifier

b. For IES2, IES2A, and (probably) IES3 instrument:

Configuration ID
DM Log Level Amplifier

SM (WIBAN1) Range
DM (WIBAN2) Range

Otherwise, the data are valid.

4. When the data are not valid set this second's cycle indicator to a null value (-1).

5. Return to the calling routine.

d. Outputs.

1. Argument list.

None.

2. COMMON blocks.

/WORKREC/

NCYCLE	-	Array of cycle indicators

e. Associated subroutines.

None.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, CYCNT1, and CYCNT2.

2. Uses COMMON block WORKREC.

2.6.8.1.2 Subroutine CYCNT1.

a. Function.

Update the running cycle counter and extract the exceptions data from the Cycle 1 ID and Configuration ID words.

b. Inputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1 minute data buffer
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. Extract the MSB and the EP mode indicator bits from the Cycle ID and save the values in the exceptions array.

2. Adjust the EP mode indicator to the Common Format value when the data are from IES.

3. If this is the first cycle processed or the previous cycle contained invalid data (BREAK is .TRUE.):

   a. Set the running cycle counter to zero.

   b. If this is not the last second of data in the current buffer, invoke VALCHK to validity check the next second of data, which should be a Cycle 2 second.

   c. If the next second of data is valid, set the cycle interrupt flag (BREAK) to .FALSE. and invoke CYCNT2 to establish the running cycle counter.

4. Adjust the running cycle counter to the proper value for the (missing) previous cycle.

5. Extract the remaining bits of information from the Cycle ID word and store the values in the exceptions block.

6. When the instrument is IES2, IES2A, or IES3, extract the necessary bits from the Configuration ID word and store the values in the exceptions block.

7. Update the running cycle counter.

   a. If the running cycle counter is 1024, reset it to 1.

   b. If the EP mode is A, B, BS, or E and the running cycle counter is 128, reset it to 1.

   c. Otherwise increment the running cycle counter by 1.

8. Return to the calling routine.

d. Outputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

None.

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT2. Process Cycle ID and Configuration ID for Cycle 2.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, and CYCNT2.

2. Uses COMMON block WORKREC.

2.6.8.1.3 Subroutine CYCNT2.

a. Function.

Update the running cycle counter and extract the exceptions data from the Cycle 2 ID and Configuration ID words.

b. Inputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
J	-	Specified second of data
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

/WORKREC/

ICUR	-	Pointer to current 1 minute data buffer
ITMUPK	-	Unpacked telemetry data buffer

3. Files.

None.

c. Processing.

1. If this is the first cycle processed or the previous cycle(s) contained invalid data (BREAK is .TRUE.):

   a. Set the running cycle counter to zero.

   b. If this is not the last second of data in the current buffer, invoke VALCHK to validity check the next second of data, which should be a Cycle 1 second.

   c. If the next second of data is valid, set the cycle interrupt flag (BREAK) to .FALSE. and invoke CYCNT1 to get the EP mode and the Cycle 1 ID MSB.

2. If this is IES instrument data, set the EP/RPA processing indicator to process both types of data.

3. If this is IES2, IES2A, or IES3 instrument data, extract the necessary bits from the Configuration ID and store the values for exceptions processing.

4. Calculate the running cycle counter from the Cycle 2 ID and the Cycle 1 MSB.

5. If the EP mode is C, D, or DS and:

   a. the running cycle counter is 0, reset it to 512,

   b. or, if the running cycle counter is 512, reset it to 1024,

   c. otherwise, if the running cycle counter is 0 reset it to 128.

6. Return to the calling routine.

d. Outputs.

1. Argument list.

BREAK	-	Data cycle interrupt indicator (LOGICAL): .TRUE. - previous cycle contained invalid data or this is the first cycle processed .FALSE. - previous cycle was valid
IXBITS	-	Array of Cycle ID and Configuration ID exceptions values
NCYCNT	-	Running cycle counter

2. COMMON blocks.

None.

e. Associated subroutines.

1. Subroutine VALCHEK. Validity check the telemetry data.

2. Subroutine CYCNT1. Process Cycle ID and Configuration ID for Cycle 1.

f. Interfaces.

1. Called by IES_XCEPT, IES2_XCEPT, IES2A_XCEPT, IES3_XCEPT, and CYCNT1.

2. Uses COMMON block WORKREC.

2.6.8.2 Subroutine IES2_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES2 instrument configuration. The specific IES2 exceptions are listed in Table 4.

All inputs, processing, outputs, associated subroutines, and interfaces are as described for Subroutine IES_XCEPT (see 2.6.8.1).

2.6.8.3 Subroutine IES2A_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES2A instrument configuration. The specific IES2A exceptions are listed in Table 5.

All inputs, processing, outputs, associated subroutines, and interfaces are as described for Subroutine IES_XCEPT (see 2.6.8.1).

2.6.8.4 Subroutine IES3_XCEPT.

a. Function.

Process telemetry data exceptions specific to the IES3 instrument configuration. The specific IES3 exceptions will be listed in Table 6.

As of May 1994, the content and format of the IES3 telemetry data has not been defined to the level of detail needed to implement the exceptions processing for the instrument. This subroutine is currently implemented as a stub and will be fully implemented when the IES3 telemetry definition is finalized.

Table 4 IES2 exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	10	2	6/all	VIP Bias
2	1	12	2,6,10...	63/all	DM LLA
3	1	13	2	11/all	Vp EL
4	1	14	2	10/all	Ion Velocity
5	1	15	2	64/all	DM Signal Level
6	1	16	2	65/all	SM (WIBAN1) Range
7	1	17	2	66/all	DM (WIBAN2) Range
8	1	145	1	1/9	Cycle1 MSB
9	1	146	1	1/6-8	EP Mode
10	1	147	2	2/5	EP/RPA Flag
11	1	148	1	1/5	Test Mode
12	1	149	1	1/4	Bias Mode
13	1	150	1	1/3	Sweep Clock
14	1	151	1	1/2	PRF/Reset
15	1	152	1	2/9	RAM/PROM
16	1	153	1	2/8	Test/Flt
17	1	154	1	2/1-7	Prog. Vers.
18	1	155	2	6/7-8	VIP Set
19	1	156	2	6/1-5	VBias Volts
20	1	158	1&2	1/	Cycle Count
1	2	8	14,30,46...	62/all	DM Electron Temp.
2	2	9	1	6/all	Vap Monitor
3	2	11	1	7/all	RPA Monitor
4	2	12	4,8,12...	63/all	DM LLB
5	2	13	1	10/all	Vp RPA
6	2	20	14,30,46...	61/all	DSM Sensor Temp.
7	2	145	1	1/9	Cycle1 MSB
8	2	146	1	1/6-8	EP Mode
9	2	147	2	2/5	EP/RPA Flag
10	2	148	2	2/7	CkSum Err.
11	2	149	2	2/6	Uplink Flag
12	2	150			Null Value
13	2	151			Null Value
14	2	152	2	2/9	RAM Err.
15	2	153	2	2/8	Dump Flag
16	2	154	2	2/1-3	Serial Number
17	2	155	2	6/7-8	VIP Set
18	2	156	2	6/1-5	VBias Volts
19	2	158	1&2		Cycle Count

Table 5 IES2A exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
1	1	4	3,7,11..	5/all	TH+
2	1	5	3,7,11..	12/all	TO+
3	1	6	1,5,9...	12/all	TE+
4	1	10	2	6/all	VIP Bias
5	1	11	1,5,9...	11/all	EP Monitor
6	1	12	2,6,10...	63/all	DM LLA
7	1	13	2,6,10...	11/all	Vp EL
8	1	15	2	64/all	DM Signal Level
9	1	16	2	65/all	SM (WIBAN1) Range
10	1	17	2	66/all	DM (WIBAN2) Range
11	1	21	1,5,9...	5/all	RPA Thermistor
12	1	145	1	1/9	Cycle1 MSB
13	1	146	1	1/6-8	EP Mode
14	1	147	2	2/5	EP/RPA Flag
15	1	148	1	1/5	Test Mode
16	1	149	1	1/4	Bias Mode
17	1	150	1	1/3	Sweep Clock
18	1	151	1	1/2	PRF/Reset
19	1	152	1	2/9	RAM/PROM
20	1	153	1	2/8	Test/Flt
21	1	154	1	2/1-7	Prog. Vers.
22	1	155	2	6/7-8	VIP Set
23	1	156	2	6/1-5	VBias Volts
24	1	158	1&2	1/	Cycle Count
1	2	4	4,8,12...	5/all	NH+
2	2	5	4,8,12...	12/all	NO+
3	2	6	2,6,10...	12/all	NE+
4	2	8	14,30,46...	62/all	DM Electron Temp.
5	2	9	1	6/all	Vap Monitor
6	2	11	3,7,11...	11/all	RPA Monitor
7	2	12	4,8,12...	63/all	DM LLB
8	2	13	4,8,12...	11/all	Vp RPA
9	2	20	14,30,46...	61/all	DSM Sensor Temp.
10	2	21	2,6,10...	5/all	EP Thermistor
11	2	145	1	1/9	Cycle1 MSB
12	2	146	1	1/6-8	EP Mode
13	2	147	2	2/5	EP/RPA Flag
14	2	148	2	2/7	CkSum Err.
15	2	149	2	2/6	Uplink Flag
16	2	150			Null Value
17	2	151			Null Value
18	2	152	2	2/9	RAM Err.
19	2	153	2	2/8	Dump Flag
20	2	154	2	2/1-3	Serial Number
21	2	155	2	6/7-8	VIP Set
22	2	156	2	6/1-5	VBias Volts
23	2	158	1&2		Cycle Count

All inputs, processing, outputs, associated subroutines, and interfaces will be as described for Subroutine IES_XCEPT (see 2.6.8.1).

Table 6 IES3 exceptions list

Num	INTO Cycle	INTO Word	FROM Cycle	FROM Word/Bits	VARIABLE NAME
				T B S	

2.6.9 Subroutine STOREM.

a. Function.

Store the unpacked telemetry data from any of the IES instrument variants into the Common Format for the APGA program.

b. Inputs.

1. Argument list.

IESTYP	-	Specified IES instrument indicator code: 1 = IES, 2 = IES2, 3 = IES2A, 4 = IES3
KOUT	-	Pointer to the 1 minute data buffer to be reformatted

2. COMMON blocks.

/WORKREC/

IPHVAL	-	Unpacked Ephemeris data buffer
ITMUPK	-	Unpacked telemetry data buffer
NCYCLE	-	Telemetry data cycle indicators buffer
ITMSEC	-	Telemetry data times buffer
IXVAL	-	Telemetry data exceptions buffer

3. Files.

None.

c. Processing.

Storage of telemetry data from the original format (ITMUPK) into the Common Format (INOLS) is table driven. Each IES variant has its own unique storage mapping (table) to control the storage from its telemetry format and exceptions list into the Common Format.

The exceptions are accessed in order during storage into the output array; i.e.: the M^{th} occurrence of the exceptions flag (MEX) in the mapping table will cause the M^{th} value from the exceptions block to be stored in the current position of the output array.

A null flag (MNV) in the mapping table indicates that the data quantity for that Common Format location does not exist in the telemetry data for the given instrument type and/or data cycle. This flag causes a null value (-1) to be stored in the corresponding position of the output array.

1. Store the Ephemeris data for the selected minute into the output buffer.

2. Establish a loop to store each second of the selected minute's Telemetry data into the output buffer, and, for each iteration of the loop:

   a. Store the Telemetry time value into the output buffer,

   b. Initialize the exceptions counter to zero,

   c. Determine the cycle type (1 or 2) for the current second,

   d. Establish a loop to store a value into each location of the output buffer (1 - 160).

      1. If the data cycle type is invalid, store a null value (-1) into the current location of the output buffer.

      2. Otherwise, retrieve the mapping value for the current location from the proper instrument/cycle map and:

         a. Store the value from the telemetry data buffer location indicated by the mapping value into the current location of the output buffer.

         or

         b. When the mapping value is the exceptions flag (MEX), increment the exceptions counter and store the value from the exceptions buffer location indicated by the exceptions counter into the current location of the output buffer.

         or

         c. When the mapping value is the null value flag (MNV), store a null value (-1) into the current location of the output buffer.

         or

         d. When the mapping value is the constants flag (MCO), store a constant value into the current location of the output buffer.

3. Return to BNBA.

d. Outputs.

1. Argument list.

   None.

2. COMMON blocks.

   /OUTREC/

IPHVAL	-	One minute block of Ephemeris data
ISECOLS	-	One minute block of telemetry data times
INOLS	-	One minute block of reformatted telemetry data

e. Associated subroutines.

None.

f. Interfaces.

1. Called by BNBA.

2. Uses COMMON blocks WORKREC and OUTREC.

2.6.10 Subroutine OUTPUT.

a. Function.

Write the reformatted RIR, Ephemeris, Telemetry time, and Telemetry data buffers to the output file.

b. Inputs.

1. Argument list.

IESPREPFIL -	Full path name for the Prepfile
IOUTUNIT -	Unit number of the output file

2. COMMON blocks.

/OUTREC/

IPHOUT	-	Reformatted Ephemeris data buffer
ISECOLS	-	Telemetry data times buffer
INOLS	-	Reformatted Telemetry data buffer

/OUTRIR/

IRIROUT	-	Reformatted RIR data buffer

c. Processing.

1. If this is the first time through, open the output file and write the RIR data buffer.

2. Write the Ephemeris, Telemetry times, and Telemetry data buffers to the output file as a single record.

3. Return to BNBA.

d. Outputs.

1. Argument list.

None.

2. Files.

PREPFILE - The file to which the current 1 minute set of reformatted data is written.

e. Associated subroutines.

None.

f. Interfaces.

1. Called by BNBA.

2. Uses COMMON blocks OUTREC and OUTRIR.

SECTION 3. ENVIRONMENT

3.1 Equipment Environment.

The SSIES Flight Data Processor (FDP) System has a major restriction mandated by the structure of the raw data Mission Sensor file received at AFSFC from AFGWC. This file consists of data packed as four 9-bit telemetry words stored in 36-bit Sperry UNIVAC words. The BNBA program reads the data into a VAX computer system and manipulates the data according to the specific constructs imposed on the input data by the VAX internal storage format. The BNBA program will require extensive modification to run properly on a system whose internal data storage conventions do not map exactly to those of the VAX.

The only other equipment constraints are that the hardware must have enough main memory to load and execute the APGA program and have enough direct access mass storage space for the various files (see Section 3.3.2 for file sizes).

3.2 Support Software.

The following system routines are required for the SSIES FDP system:

a. Program APGA.

   1. VAX intrinsic functions: IDATE, TIME

b. Program BNBA.

   1. VAX intrinsic functions: IBITS, MVBITS

c. Program LDCON02.

   None.

3.3 Data Base.

The data base file system consists of multiple files; some are used exclusively as inputs, some are output by one processing program and input by another, and some are exclusively output files.

3.3.1 General Characteristics.

3.3.1.1 Input Data Base.

3.3.1.1.1 SSIES Mission Sensor Raw Data File.

a. Name: TBS by AFSFC.

b. Usage: Transmitted to AFSFC from AFGWC. This is the input file for the BNBA data unpacking and reformatting program.

c. Data Permanence: TBS by AFSFC.

d. Storage: This file is located on a mass storage disk. The record and file sizes are variable.

e. Access type: INDEX/KEYED

f. Format: BINARY

g. Restrictions: The BNDA program records this file. The BNBA program uses the file on a read-only basis.

3.3.1.1.2 SSIES Common Format Data File.

a. Name: IESPREPFILE.

b. Usage: Produced by the BNBA program and used as the primary input to the APGA program.

c. Data Permanence: TBD by AFSFC.

d. Storage: This file is located on a mass storage disk. The size of this file is variable, but a nominal estimate of size is 500,000 words for 101 fixed length records of 4888 words each. This estimate approximates the amount of data in a single readout REV.

e. Access Type: SEQUENTIAL

f. Format: BINARY

g. Restrictions: None.

3.3.1.2 Output Data Base.

3.3.1.2.1 Environmental Data Record File.

a. Name: IESEDRFILE.

b. Usage: Loaded and controlled by the APGA program.

c. Data Permanence: TBD by AFSFC.

d. Storage: This file is located on a mass storage disk. The file consists of a variable number of 1791 word fixed length records.

e. Access type: DIRECT

f. Format: BINARY

g. Restrictions: None.

3.3.1.2.2 Statistical Information File.

a. Name: IESSTATFILE.

b. Usage: This file is created by the APGA program.

c. Data Permanence: The data in this file are perishable.

d. Storage: This file is located on a mass storage disk. The file consists of a variable number of 833 word fixed length records.

e. Access type: DIRECT

f. Format: BINARY

g. Restrictions: None.

3.3.1.2.3 AGDB Transfer Files.

a. Name: IESAGDBXFR1 (and IESAGDBXFR2).

b. Usage: These files are created the APGA program.

c. Data Permanence: The data in these files are perishable.

d. Storage: These files are located on a direct access mass storage disk. The files consist of a variable number of 1791 word fixed length records.

e. Access type: DIRECT

f. Format: BINARY

g. Restrictions: None.

3.3.1.3 Control/Information Data Base.

3.3.1.3.1 APGA Program Control file.

a. Name: IESCNTRLFILE.

b. Usage: This file is built using program LDCON02 and accessed by the APGA program.

c. Data Permanence: This is a quasi-permanent file which is modified only when one of the control parameters in the file needs to be updated or a new satellite is added.

d. Storage: This file is located on a mass storage disk. The file size is 1095 words and contains control parameters for a maximum of three DMSP spacecraft.

e. Access type: SEQUENTIAL

f. Format: BINARY

g. Restrictions: None.

3.3.1.3.2 Apex Coordinate Conversion File.

a. Name: IESAPEXTABLE.

b. Usage: This file is accessed by the APGA program.

c. Data Permanence: This is a permanent file which should not be changed.

d. Storage: This file is located on a mass storage disk. The file size is 8203 words.

e. Access type: SEQUENTIAL

f. Format: BINARY

g. Restrictions: None.

3.3.2 Organization and Detailed Description.

3.3.2.1 SSIES Mission Sensor Raw Data File.

a. Contents.

This file contains packed vehicle, orbit, readout REV, Ephemeris, and telemetry data as downlinked from a DMSP SSIES satellite. The data are packed in the Sperry 9 bit byte and 36 bit word convention.

b. Size:

This is a variable size file. The actual size is dependent upon the number of minutes of data contained in the file, and the telemetry configuration of the particular satellite from which the data originated.

c. Format:

Binary index/keyed
I/O record size - variable (instrument dependent)
Data record size - variable (instrument dependent)

d. Structure:

The file consists of a Readout Information Record (RIR), which appears once as the first record of the file, and an alternating sequence of ephemeris and instrument telemetry data records. The exact content and format of the data in these files are described in sections 3.3.2.2.4 - 6 of the AFGWC and SFC Interface Specification.

3.3.2.2 File IESPREPFILE.

A detailed description of this file and its contents can be found in Volume 2, Programmer's Guide, Appendix 8.3. The APGA program reads data from this file in one minute records. Each record contains an ephemeris data block and 60 one second data frames containing the unpacked SSIES sensor data. Table 7 lists the ephemeris data actually used by the APGA program, and Figure 22 and

Table 8 and Table 9 describe the format and content of the Common Format 1 second data frame.

Table 7 DMSP ephemeris block data used by program APGA

Word	Item	Units
1	Latitude 1	Radians
2	Longitude 1	Radians
4	Julian day 1	Day-of-year
5	Time 1	Seconds since midnight
6	Latitude 2	Radians
7	Longitude 2	Radians
9	Julian day 2	Day-of-year
10	Time 2	Seconds since midnight
19	Altitude 1	Nautical miles
22	Altitude 2	Nautical miles
23	Orbit angle 1	Radians
24	Orbit angle 2	Radians

Notes:

1. Labels 1 and 2 in the table refer to the first and second satellite locations in the ephemeris record. Location 1 is the location of the satellite at the time of the latest (in time) data frame in the one minute block. Location 2 is the location of the satellite at the time of the earliest (in time) data frame in the one minute block. Time 2 is nominally 60 seconds prior to Time 1.

2. The orbit angle is the angular distance to the last ascending node along the orbit track.

Word Number	Word Identification											
1 - 12	Cycle 1 ID	Config 1 ID	OLS Cmd Mon	MP Temp:H+	MP Temp:O+	MP Temp:e	Current Monitor	Elmeter Temp	Vaper Monitor	Bias Monitor	Electron Monitor	DM LLA
	Cycle 2 ID	Config 2 ID	DSM Cmd Mon	MP Dens:H+	MP Dens:O+	MP Dens:e	ADC Temp Monitor	DM Temp Electron			RPA Ion Monitor	DM LLB
13 - 24	MP EP Veh Pot	MP Ram Velocity	DM Signal Level	SM WIBAN1 Range	DM WIBAN2 Range	Not Used	Not Used	Not Used	RPA Therm	DSM: EL/AMP	-->	-->
	MP RPA Veh Pot					Subcom 1	Subcom 2	DSM Sens Temp	EP Therm			
25 - 36	DSM: EL/AMP	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
37 - 48	DSM: ELE/AMP	-->	-->	-->	-->	-->	-->	-->	-->	DSM: DRIFT	-->	-->
49 - 60	DSM: DRIFT	-->	-->	-->	-->	-->	-->	-->	-->	RPA Current	-->	-->
61 - 72	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
73 - 84	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->

Figure 22 Common Format data frame (part 1)

Word Number	Word Identification											
85 - 96	RPA Current	-->	-->	-->	-->	-->	-->	-->	-->	EP Current	-->	-->
97 - 108	EP Current	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->	-->
109 - 120	EP Current	-->	-->	-->	-->	-->	-->	-->	-->	SMFIBA1	SMFIBA2	SMFIBA3
121 - 132	SMFIBA4	SMFIBA5	SMFIBA6	SMFIBA7	SMFIBA8	SMFIBA9	DMFIBA1 / Not Used	DMIFBA2 / Not Used	DMFIBA3 / Not Used	DMFIBA4 / Not Used	DMFIBA5 / Not Used	DMFIBA6 / Not Used
133 - 144	EP Sweep Monitor	-->	-->	-->	-->	-->	RPA Sweep Monitor	-->	-->	-->	-->	-->
145 - 156	Cycle 1 MSB	EP Mode	EP/RPA Flag	Test Mode / Checksum	Bias Mode / Uplink Flag	Sweep Clock / Not Used	PRF Reset / VBias Monitor	RAM/PROM Op. / RAM Error	Test/Flt Flag / Dump Flag	Prog. Version / Serial Number	VIP Setting	VBias Volts
157 - 160	Inst. Code	Cycle Count	Not Used	Not Used								

All cycles	Identification of telemetry item appearing in both even and odd cycles
Odd Cycles	Identification of telemetry item appearing in odd cycles
Even Cycles	Identification of telemetry item appearing in even cycles
Not Used	Spare word. Contains -1.

Figure 22 Common Format data frame (part 2)

Table 8 Common Format data words (Cycle 1)

	CYCLE 1	
Acronym	Word	Description
Cycle ID	1	Cycle identifier
Configuration ID	2	Configuration bits
OLS Command	3	Last command sent from OLS
T (H+)	4	H+ Temperature calculated by microprocessor
T (O+)	5	O+ Temperature calculated by microprocessor
T (E)	6	Electron temperature calculated by microprocessor
Current Monitor	7	Current (+28 V) Input from spacecraft
MEP Temp 1	8	Temperature at MEP Electrometer
V(ap) Monitor	9	Measured voltage on Ion Array
Vbias + Vip Monitor	10	Setting levels for V(BIAS) + VIP
EP Monitor	11	Flags from microprocessor EP analysis
DM LLA	12	Output of Drift Meter log amplifier A
VP-EP	13	Vehicle potential from calculation of EP data
Vsp	14	Calculated down-range relative plasma velocity
DM Signal level	15	Drift meter zero level
SM Range (WIBAN1)	16	SM Electrometer/Amplifier range setting indicator
DM Range (WIBAN2)	17	Drift meter range indicator
Unused (NULL)	18	Spare location
Unused (NULL)	19	Spare location
Unused (NULL)	20	Spare location
RPA Temperature	21	Temperature from Thermistor on back of RPA sensor housing
SM data	22 - 45	Total Ion Trap (SM) Electrometer/Amplifier output
DM Data	46 - 57	Drift Meter offset output
RPA Current Data	58 - 93	Ion RPA log amplifier output of current to collector
EP Current Data	94 - 117	Electron Probe log amplifier output of current to collector
SM Filter	118 - 126	Output from filters connected to SM Collector
DM Filter (FIBA)	127 - 132	Output from filters connected to DM Collector
EP Sweep Monitor	133 - 138	Voltage applied to Electron retarding grid
RPA Sweep Monitor	139 - 144	Voltage applied to Ion RPA retarding grid
Cycle 1 MSB	145	Most Significant Bit of Cycle ID
EP Mode	146	Electron Probe operation mode indicator (0 - 6 = A - E)
EP / RPA Flag	147	Indicates whether EP, RPA, or both types are in telemetry
Test Mode Flag	148	0: OFF, 1: ON
Bias Mode Flag	149	1 for 8 second EP bias setting sweeps, otherwise 0
Sweep Clock Flag	150	1 for normal EP and RPA operation (sweep clocks on)
PRF/Reset Flag	151	1 indicates that program will restart on next cycle
RAM / PROM Flag	152	Microprocessor program source 0: PROM, 1: RAM
Test/Flight Mode Flag	153	Microprocessor program mode 0: Flight, 1: Test
Program Version	154	5: program stored in PROM 6: modified output format
Vip Monitor	155	Vip Setting
Vbias Monitor	156	Vbias setting
Instrument Code	157	Source of raw telemetry data 1: IES, 2: IES2, 3: IES2A, 4: IES3
Cycle Count	158	Running cycle counter (1-1024)
Unused (NULL)	158 - 160	Spare locations

Table 8 Common Format data words (Cycle 2)

	CYCLE 2	
Acronym	Word	Description
Cycle ID	1	Cycle identifier
Configuration ID	2	Configuration bits
DSM Command Monitor	3	Last command sent from DSM
N (H+)	4	H+ Density calculated by microprocessor
N (O+)	5	O+ Density calculated by microprocessor
N (E)	6	Electron Density calculated by microprocessor
MEP Temp 2	7	Temperature at MEP ADC
DM Temp	8	DM Electronics Temperature
V(ap) Monitor	9	Measured voltage on Ion Array
Vbias + Vip Monitor	10	Setting levels for V(BIAS) + VIP
RPA Monitor	11	Flags from microprocessor EP analysis
DM LLB	12	Output of Drift Meter log amplifier B
VP-RPA	13	Vehicle potential from calculation of RPA data
Vsp	14	Calculated down-range relative Ion plasma velocity
DM Signal level	15	Drift meter zero level
SM Range (WIBAN1)	16	SM Electrometer/Amplifier range setting indicator
DM Range (WIBAN2)	17	Drift meter range indicator
Subcom 1	18	DSM Multiplex data 1
Subcom 2	19	DSM Multiplex data 2
DSM Sensor Temp	20	Temperature from DSM sensor
EP Temperature	21	Temperature from Thermistor on EP pre-amp housing
SM data	22 - 45	Total Ion Trap (SM) Electrometer/Amplifier output
DM Data	46 - 57	Drift Meter offset output
RPA Current Data	58 - 93	Ion RPA log amplifier output of current to collector
EP Current Data	94 - 117	Electron Probe log amplifier output of current to collector
SM Filter	118 - 126	Output from filters connected to SM Collector
DM Filter (FIBA)	127 - 132	Output from filters connected to DM Collector
EP Sweep Monitor	133 - 138	Voltage applied to Electron retarding grid
RPA Sweep Monitor	139 - 144	Voltage applied to Ion RPA retarding grid
Cycle 1 MSB	145	Most Significant Bit of Cycle ID
EP Mode	146	Electron Probe operation mode indicator (0 - 6 = A - E)
EP / RPA Flag	147	Indicates whether EP, RPA, or both types are in telemetry
Checksum Error Flag	148	0: None, 1: ERROR
Uplink Flag	149	1: program has stopped to receive uplink
Unused (NULL)	150	Spare location
Vbias/Senpot Flag	151	0: Vbias Mode 1: Senpot Mode
RAM Error Flag	152	1: RAM Error has occurred
Dump Flag	153	1: Program is dumping RAM
Serial Number	154	SSIES serial number
Vip Monitor	155	Vip Setting
Vbias Monitor	156	Vbias setting
Instrument Code	157	Source of raw telemetry data 1: IES, 2: IES2, 3: IES2A, 4: IES3
Cycle Count	158	Running cycle counter (1-1024)
Unused (NULL)	158 - 160	Spare locations

3.3.2.3 File IESEDRFILE.

a. Contents:

This file contains the Environmental Data Records generated by the APGA program, and also includes processing status flags used by the APGA program.

b. Size:

The size of this file is determined by the time span of the data which comprise the file. This can be calculated from:

$$wEDR = [NMin / 3] \times RSize$$

where: wEDR is the number of words of data in the file,
NMin is the number of minutes of data,
RSize is the record size (1791 words).

c. Format:

Binary direct access
I/O record size - fixed length, 1791 words
Data record size - 595 words

d. Structure:

The file consists of a header record, unnecessary for AFSFC purposes, but containing data required by the Sperry database system, and a series of physical data records. Each physical data record consists of a (now obsolete) 6 word status flag block and three 595 word logical data records. Each logical data record constitutes one minute's worth of data.

word#	contents
1	Mission ID (4-digit integer)
2	Date of oldest data in file (YYMMDD, integer)
3	Time of oldest data in file (HHMMSS, integer)
4	Time of oldest data (IES minutes, integer)
5	Record number of oldest data in file (integer)(relative to header block)
6	Date of latest data in file (YYMMDD, integer)
7	Time of latest data in file (HHMMSS, integer)
8	Time of latest data (IES minutes, integer)
9	Record number of latest data in file (integer)(relative to header block)
10	Current track size of IESEDRFILE (integer)
11	Number of active satellites (integer)
12-1611	Processing status flags (two per EDR I/O block)
1612-1791	[Zero fill]

Figure 23a Structure and contents of header blocks in file IESEDRFILE

Figure 23a shows the structure of the header block. Note that most of the information is useless for AFSFC processing, but is filled with innocuous values by the APGA program for processing. The

entire EDR I/O processing should be simplified to eliminate the obsolete complications imposed upon it by the AFGWC Sperry database system.

Figure 23b shows the structure of an I/O record. The first six words in each I/O record were processing status flags used by the Sperry database system, but useless in the AFSFC scheme. The APGA program fills these locations with innocuous values for processing. The internal structure of an EDR is illustrated in Volume 2, Programmer's Guide, Appendix 8.6.

word#	contents
1-6	Processing status flags (two per EDR in block)
7-601	EDR #1
602-1196	EDR #2
1197-1791	EDR #3

Figure 23b Structure of I/O records in file IESEDRFILE

3.3.2.4 File IESSTATFILE.

a. Contents:

This file contains data processing statistics from an APGA program run. Included are readout-averaged values for various geophysical parameters and instrument temperatures, processing statistics for RPA and EP sweep analyses and C_kL calculations, and error/status flags for each processing module.

b. Size:

The file size is 833 words, which provides room for statistics from 18 readout REVs. This is a holdover from AFGWC Sperry processing. This file is now initialized at the start of each APGA program execution, and will contain information only for the satellite and rev of the current run. The input and output processing for this file could be greatly simplified or possibly eliminated for AFSFC purposes.

c. Format:

Binary direct access
I/O record size = 833 words
Data record size = 46 words

d. Structure.

Each file is an single I/O record (833 words) which contains a 4 word header and a 46 word data record. Figure 24 shows the structure and contents of a data record.

3.3.2.5 File IESAGDBXFR1 (and IESAGDBXFR2).

a. Contents:

These files, generated by the APGA program, contain one minute summary records created from

Word	Contents (All entries are INTEGER unless labeled otherwise)
1	Mission ID
2	Satellite flight ID
3	Time file last displayed (IES minute)
4	Time file last updated (IES minute)
5	Readout REV number
6	Readout REV start date (DD)
7	Readout REV start time (HHMM)
8	Average EP N_e (el/cm^3)
9	N_e analysis source 1 - Ground analysis 2 - On-board microprocessor analysis
10	EP sensor mode (0-6 : A-E)
11	Average RPA N_i (ion/cm^3)
12	N_i analysis source 1 - Ground analysis 2 - On-board microprocessor analysis
13	Average DM N_i (ion/cm^3)
14	Average SM N_i (ion/cm^3)
15	Average EP T_e (°K)
16	Average RPA T_i (°K)
17	Average of first SM filter PDS (REAL)
18	Average C_kL (REAL)
19	C_kL data source 1 - SM density data only 2 - SM density and filter data 3 - EP DC mode density data only
20	Average horizontal cross-track ion drift velocity (m/s)
21	Average vertical ion drift velocity (m/s)
22	Average horizontal ram drift velocity (m/s)
23	Average V_{IP}(volts) (REAL)
24	Average V_{bias}(volts) (REAL)
25	V_{bias}source 1 - On-board microprocessor analyses 2 - SENPOT sensor
26	V_{EP}(volts) (REAL)
27	Average electrometer instrument temperature (°C)
28	Average ADC temperature (°C)
29	Average DSM instrument temperature (°C)
30	Average DM offset voltage (volts) (REAL)
31	Number of successful EP sweep analyses
32	Number of EP sweeps attempted
33	Number of successful RPA sweep analyses
34	Number of RPA sweeps attempted
35	Number of successful C_kL calculations
36	Number of C_kL calculations attempted
37	EP status/error flag
38	RPA status/error flag
39	DM status/error flag
40	SM status/error flag
41	MP status/error flag
42	C_kL status/error flag
43	Duplicate data frame count
44	Average difference, SM to DM density
45	Average difference, SM to EP density
46	Average difference, SM to RPA density
47	Average difference, DM to EP density
48	Average difference, DM to RPA density
49	Average difference, EP to RPA density
50	Number of $T_e < T_i$ points

Figure 24 Structure and contents of file IESSTATFILE

EDRs. They are a remnant of AFGWC Sperry processing and are probably not necessary for AFSFC purposes. File IESAGDBXFR2 has been disabled to somewhat simplify processing for AFSFC.

b. Size:

The file contains a header and a variable number data records.

c. Format:

Binary direct access
I/O record size - 1791 words
Data record size - 22 words

d. Structure:

The first I/O record contains a 22-word header block and 80 data records. Each subsequent I/O record contains 81 data records. The records are output to the file in the order the data were processed by the APGA program. Figure 25a shows the structure of these files, and Figure 25b shows the structure and contents of a data record.

3.3.2.6 File IESCNTRLFILE.

a. Contents:

This file contains program control information, SSIES sensor constants, and algorithm control parameters used by program APGA. The file is loaded by program LDCON02. (Note: A detailed description of the various parameters in this file can be found in Volume 2, Programmer's Guide, Appendix 8.4)

b. Size:

The file size is 1095 words.

c. Format:

Binary direct access
I/O record size = 1095 words

d. Structure.

The file is divided into a header section and up to three control information sections, one per active satellite.

3.3.2.7 File IESAPEXTABLE.

a. Contents:

This file contains information required to convert geographic latitude and longitude to modified Apex latitude and longitude. (Note: The conversion is valid for an altitude of 840 kilometers.)

a. I/O Record #1.

word#	contents
1-22	Header information 1 - File lock word (99 for lock) 2 - Date file built by APGA program (YYMMDD) 3 - Time file built by APGA program (HHMMSS) 4 - Number of satellites in file (1-8) 5 - Number of records in file (total) 6 - Satellite 1 flight ID 7 - Number of records for satellite 1 8 - Satellite 2 flight ID 9 - Number of records for satellite 2 - - - 20 - Satellite 8 flight ID 21 - Number of records for satellite 8 22 - [Zero fill]
23-44	Data record #1 (for one minute of data)
45-66	Data record #2
-	-
-	-
-	-
1760-1782	Data record #80
1783-1791	[Zero fill]

b. I/O Record #2-N

word#	contents
1-22	Data record #1
23-44	Data record #2
-	-
-	-
-	-
1760-1782	Data record #81
1783-1791	[Zero fill]

Figure 25a Structure of files IESAGDBXFR1 (and IESAGDBXFR2)

word#	contents
1	Date (YYMMDD, integer)
2	Time (HHMM, integer)
3	Data used in C_kL calculation 1 - SM density data only 2 - SM density and filter data 3 - EP DC mode density data only
4	Geographic latitude (degrees, north)
5	Geographic longitude (degrees, east)
6	Electron/ion density (part/cm^3)
7	Electron/ion density source (integer) 1 - SM (ion density) 2 - DM (ion density) 3 - EP (DC modes) (electron density) 4 - EP (sweeps) (electron density) 5 - RPA (ion density)
8	Ion constituent ratio, N[H+]/N[O+]
9	Electron temperature (°K)
10	Ion temperature (°K)
11	C_kL for first 10-second period (hhmm00-hhmm09) >0: C_kL value -1: No C_kL calculation possible -2: (RMS ΔN)/N below threshold
12	p_1 for first 10-second period <0: Calculation was from 256-point data set >0: Calculation was from 512-point data set
13	C_kL for second 10-second period (hhmm10-hhmm19)
14	p_1 for second 10-second period
-	-
-	-
-	-
21	C_kL for sixth 10-second period (hhmm50-hhmm59)
22	p_1 for sixth 10-second period

Figure 25b Structure and contents of data records from files IESAGDBXFR1 (and IESAGDBXFR2)

b. Size:

The file size is 8203 words.

c. Format:

Binary direct access
I/O record size = 8203 words

d. Structure:

The structure and contents of this file are shown in Figure 26.

a. File structure.

word#	contents
1	Valid altitude for coordinate system (km)
2-4	Geocentric Cartesian coordinates of the center of the offset-dipole coordinate system in which the Apex look-up table is defined.
5-13	Rotation matrix for conversion between geographic and offset-dipole coordinate systems, U(3,3).
14-8203	Offset-dipole-to-Apex coordinate transformation look-up table, APXTAB(2,45,91).

b. Look-up table structure.

The first index in the look-up table array, APXTAB(I,J,K), indicates whether Apex latitude (I=1) or longitude (I=2) is being accessed. The second index is the longitude index, which is calculated from:

$$J = (\text{offset-dipole east longitude})/8 + 1.$$

The third index is the latitude index, which is calculated from:

$$K = (\text{offset-dipole latitude})/2 + 46.$$

For example, the Apex latitude for offset-dipole latitude 68° south, longitude 24° west (336° east) would be located in APXTAB(1,43,12).

Figure 26 IESAPEXTABLE structure and contents

SECTION 4. PROGRAM MAINTENANCE PROCEDURES

4.1 Conventions.

4.1.1 COMMON Block Names.

a. xxxCON

COMMON blocks of this form contain the processor control information and sensor constants for the various instruments. The data which reside in these COMMON blocks are read from the IESCNTRLFILE and stored into the COMMON blocks during program initialization. See Subroutines INIT (2.4.2) and VEHPAR (2.4.2.4) for specifics. The xxx indicates the instrument or processor with a two or three letter mnemonic:

PRO	- Processing Control
RPA	- Retarding Potential Analyzer
EP	- Electron Probe
SM	- Scintillation Monitor
DM	- Drift Meter
CKL	- C_kL Processor
DF	- Data Frame Decode Processor
MP	- Onboard MicroProcessor

b. xxxANL

COMMON blocks of this form contain the results of the analysis for each of the various sensors. The xxx will indicate the instrument or processor as in 4.1.1a above.

4.1.2 Variable Names.

To prevent naming conflict and to simplify identification of the data and processors involved in any segment of code, most important variables are named according to the various instruments.

Instrument	REAL variable	INTEGER variable
RPA	RPxxxx	IRPxxx
EP	EPxxxx	IEPxxx
SM	SMxxxx	ISMxxx
DM	DMxxxx	IDMxxx
MP	xxxxMP	MPxxxx

4.1.3 Coordinate Systems.

4.1.3.1 Geocentric Coordinate Systems.

The APGA program employs three geocentric spherical (latitude, longitude) coordinate systems: geographic, offset-dipole, and modified Apex. These systems are defined as follows:

a. Geographic latitude/longitude.

The geographic system is the standard latitude/longitude system with north latitudes positive and east longitudes positive.

b. Offset-dipole latitude/longitude.

Offset-dipole coordinates are defined by a dipole which has been tilted and offset so that the poles of the dipole are co-located with the poles of the modified Apex system (see reference 1.2.3o). This coordinate system is used solely in the conversion from geographic coordinates to modified Apex coordinates. The transformation from geographic to offset-dipole coordinates is done in four steps:

1. Convert the geographic latitude and longitude to a geocentric Cartesian coordinate system for an altitude of 840 km.

2. Apply a pre-defined rotation matrix to the Cartesian coordinates to rotate and tilt the axes of the coordinate system.

3. Subtract offset values from the Cartesian coordinates to move the center of the coordinate system.

4. The rotated/offset Cartesian coordinates are then converted back to a spherical system to provide offset-dipole latitude and longitude.

This transformation is made using the rotation matrix and offset definitions for the offset-dipole system stored in file IESAPEXTABLE. This algorithm is implemented in Subroutine OFFSET (see 2.4.3.2.2).

c. Modified Apex latitude/longitude.

Modified Apex coordinates are a geomagnetic latitude/longitude system based on a "real-field" model of the earth's magnetic field. A description of Apex coordinates can be found in reference 1.2.3u, a copy of which is in the SSIES Project Workbook. Briefly, Apex coordinates are nearly identical to invariant and corrected geomagnetic coordinates at high latitudes and are nearly identical to dip latitude at low latitudes. The Apex coordinates used in this application are "modified" in that they have been adjusted so that the latitudes produced by the coordinate transformation pass through 0° at the magnetic (dip) equator, rather than discontinuously jumping from a minimum positive latitude to the corresponding negative latitude at the equator. This modification is described in more detail in reference 1.2.3o. For the sake of brevity, modified Apex latitude and longitude are denoted simply Apex latitude and longitude in this document. As with the offset-dipole coordinate system, Apex coordinates are a function of the altitude above the earth's surface.

The transformation from geographic to Apex coordinates is done in two steps. First, the geographic latitude and longitude are converted to offset-dipole coordinates. Apex coordinates are derived from the offset-dipole coordinates by quadratic interpolation in a large look-up table, also stored in file IESAPEXTABLE. This table consists of the Apex latitude and longitude for every point on an 8° latitude by 15° longitude offset-dipole grid. This algorithm is implemented in Subroutines APXTAB (see 2.4.3.2.1) and INTERP (see 2.4.3.2.3). This table was generated using software described in reference 1.2.3o and was generated using the IGRF 1990 standard geomagnetic field model initialized to epoch 1990. Both offset-dipole and Apex coordinates were

calculated for an altitude of 840 km. Both geomagnetic systems are oriented in the same sense as the geographic system, i.e., north latitudes and east longitudes are positive.

4.1.3.2 Spacecraft-Centered Coordinate Systems.

The APGA program employs four spacecraft centered Cartesian coordinate systems: spacecraft coordinates (x_s, y_s, z_s), geographic meridian coordinates (x_g, y_g, z_g), geomagnetic meridian coordinates (x_m, y_m, z_m), and irregularity coordinates (x_i, y_i, z_i). These systems are defined as follows:

a. Spacecraft coordinates.

This coordinate system is tied to the orbital orientation of the spacecraft with $+y_s$ in the direction of travel along the orbit, $+x_s$ in the nadir direction, and $+z_s$ in the direction of the solar panel. This is the same system used in other DMSP documentation. The drift velocity observations reported from the DM and RPA sensors are defined in this system so that the total ion drift velocity with respect to the spacecraft is given by

$$\underline{u} = u_v x_s + u_r y_s + u_h z_s$$

b. Geographic-meridian coordinates.

This coordinate system is tied to the orientation of the spacecraft in the geocentric geographic system with $+x_g$ in the plane of the local geographic meridian perpendicular to the local nadir direction in the local north direction, $+y_g$ in the local east direction also perpendicular to the local nadir direction, and $+z_g$ in the direction of the local nadir. (Note that $+z_g$ is in the same direction as $+x_s$.) The transformation from spacecraft coordinates to geographic meridian coordinates is given by:

$$\begin{vmatrix} x_g \\ y_g \\ z_g \end{vmatrix} = \begin{vmatrix} 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \\ 1 & 0 & 0 \end{vmatrix} \begin{vmatrix} x_s \\ y_s \\ z_s \end{vmatrix}$$

where α is the rotation angle between the direction of orbital motion and the geographic meridian given by:

$$\sin\alpha = \frac{\cos i}{\cos \lambda_g}$$

and i is the orbital inclination angle,
λ_g is the local geographic latitude.

This transformation is implemented in Subroutine CTRANS (see 2.4.4.6.16). (Note: The transformation from geographic meridian coordinates to spacecraft coordinates can be calculated using the inverse of the above matrix equation.)

c. Geomagnetic meridian coordinates.

This coordinate system is similar to the geographic meridian coordinate system, except that $+x_m$ and $+y_m$ are oriented with respect to the local geomagnetic meridian (as specified by the local

magnetic declination angle) in the magnetic north and east direction, respectively. The $+z_m$ coordinate is in the same direction as $+z_g$. The transformation from geographic meridian to geomagnetic meridian coordinates is given by:

$$\begin{vmatrix} x_m \\ y_m \\ z_m \end{vmatrix} = \begin{vmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_g \\ y_g \\ z_g \end{vmatrix}$$

where: β is the local magnetic declination angle.

This transformation is made in Subroutine CVEFF (see 2.4.4.6.15). (Note: As with the previous transformation, the reverse transformation can be calculated from the inverse of the forward transformation matrix.)

d. Irregularity coordinates.

This coordinate system is aligned with the coordinate system used to define small-scale irregularities in the ionospheric plasma. In practice, this system is aligned with the geomagnetic-meridian system described above, although in theory it may not be if the irregularities are not aligned with the local geomagnetic field direction. The scaling of this coordinate system is defined by the axial ratios of the irregularities, parameters a and b, which are provided from a sub-set of the ionospheric irregularity model from the WBMOD scintillation program (see reference 1.2.3q). A description of this coordinate system and the transformation equations (used in calculating the effective satellite velocity) can be found in reference 1.2.3v, a copy of which can be found in the SSIES Project Workbook (see the C_kL section).

These coordinate systems are used explicitly only in the C_kL calculation in which the velocity of the spacecraft with respect to the small scale irregularities in the irregularity coordinate system is needed to calculate C_k from the spectral parameters T and p. The use of spacecraft coordinates is implicit in the two orientation and sign parameters stored in file IESCNTRLFILE which are used to calculate u_h and u_v from the DM offset voltages (see 2.4.4.4 and Appendix B.3).

4.1.3.3 Time Systems.

The APGA program employs four time systems: spacecraft time, Greenwich Mean Time (GMT or UT), IES minutes (database time), and Apex (geomagnetic) local time. These systems are defined as follows:

a. Spacecraft time.

This time is obtained from the data frame records input from file IESPREPFILE. It is nominally the number of seconds from the previous GMT midnight, although the clock is occasionally not reset exactly at midnight so it can be greater than 86,400. The program uses this time for all data-frame related operations.

b. GMT.

This time is calculated from the spacecraft time. Corrections are made for the midnight reset problem, and the seconds since midnight are converted to integer HHMMSS where HH is the hour, MM is the minute, and SS is the second. GMT is used only for output purposes.

c. IES minutes.

This time system is used for management of the APGA database files. It is defined as the number of minutes since 1 January 1985 (an arbitrary selection) and as such is similar in concept to the AFGWC Julian hours. It provides a "linear" time coordinate which mitigates the problems associated with the ends of days, months, and years and other calendar-related oddities such as leap years. The conversion between GMT and IES minutes is implemented in Subroutine TIMCON (see 2.4.8.12).

d. Apex local time.

This time system is the local time associated with the Apex coordinate system. It is calculated in a similar manner to local (mean) solar time in that it is defined in terms of the difference between the local longitude and the longitude of the mean sun. This is done by calculating the Apex longitude of the mean anti-solar point (i.e., the Apex longitude of the point on the geographic equator at a longitude 180° from the location of the sun specified by the current GMT), and dividing the difference between the local Apex longitude and the longitude of the mean anti-solar point by 15°. This provides the Apex local time in decimal hours. This time is used both for output (in EDRs) and in calculating ionospheric model parameters used in the C_kL calculation.

4.1.4 Data Flow Control Flags

There are five data flow control flags set by the data input routines which control the processing of data by the APGA program. These control flags indicate the following:

a. NEWSAT.

This flag is always set to .TRUE. for the first data frame processed in any program run. It will subsequently be set to .TRUE., when input processing encounters data from a different satellite than the last data frame processed in the current program run, and will cause the APGA program to terminate. This flag resides in COMMON block PROCON.

b. NEWREV.

This flag is always set to .TRUE. for the first data frame processed in any program run. It will subsequently be set to .TRUE., when input processing encounters data from a different readout REV than the last data frame processed in the current program run, and will cause the APGA program to terminate. This flag resides in COMMON block PROCON.

c. GRESET.

When this flag is set to .TRUE., it indicates that there has been an acceptable time jump between the current data frame and the last data frame processed. This allows the various modules to reset themselves for a time jump without a full reinitialization. This flag resides in COMMON block PROCON.

d. THEEND.

When this flag is set to .TRUE., it indicates that the current data frame is the end of data for the current run of the APGA program.

e. IQFRAME.

This variable, set in Subroutine CHEKIT (see 2.4.3.3), indicates the state of the data in the current frame. The settings are as follows:

-2: - Do not process / Data frame time discontinuity.
-1: - Do not process / Duplicate frame.
0: - Do not process / Frame data not valid.
+1: - Process / Frame data is valid.

4.2 Verification Procedures.

Changes to the APGA program should be verified via regression testing against the established baseline code and data set. If the modifications of the APGA program are such that the output should not change, then the results of the regression test should be identical to those obtained from the baseline run(s). If the modifications will change the output, the results of the regression test should be identical for those sections of the program not changed.

4.3 Error Conditions.

The various error conditions which programs APGA, BNBA, and LDCON02 may encounter are described in detail in Volume 2, Programmer's Guide (see Sections 3.4.2, 4.4.2, and 4.5).

4.4 Special Maintenance Procedures.

4.4.1 Program Compilation and Collection.

The programs which make up the SSIES processing system are compiled using VAX DCL commands as follows:

a. Program APGA.

```
$ FOR/LIST/SHO=(INCLUDE,NOMAP) APGA.FOR
$ LINK APGA.OBJ
$ DELETE/NOCONFIRM APGA.OBJ
```

b. Program BNBA.

To configure BNBA for SSIES-2, use the following logical assignment:
```
$ DEFINE/USER SENSNAM SENSNAM_2.INC
```

To configure BNBA for SSIES-2A, use the following alternative logical assignment:
```
$ DEFINE/USER SENSNAM SENSNAM_2A.INC
```

```
$ FOR/LIST/SHO=(INCLUDE,NOMAP) BNBA.FOR
$ LINK BNBA.OBJ
$ DELETE/NOCONFIRM BNBA.OBJ
```

c. Program LDCON02.

```
$ FOR/LIST/SHO=(INCLUDE,NOMAP) LDCON02.FOR
$ LINK LDCON02.OBJ
```

$ DELETE/NOCONFIRM LDCON02.OBJ

4.4.2 Modification of Parameters in IESCNTRLFILE.

The various control parameters in file IESCNTRLFILE are set via program LDCON02. Detailed instructions for the use of this program are given in Section 8.4 of Volume 2, Programmer's Guide.

4.4.3 Limiting Amount of Data Processed.

The amount of data processed by the APGA program can be constrained by setting up a file containing the data (day-of-year) and time (seconds since midnight) at which to start processing and the number of seconds of data to process. This file, named IESPROLIMITS, should be an ASCII formatted data file containing one line of data with the above information in the order stated. If the APGA program cannot locate this file (as is the case in a normal production run) or encounters an error in reading from the file, all data in file IESPREPFILE will be processed. If the file is successfully read, the APGA program will process no data input from file IESPREPFILE until a one-minute data block is input in which the time of the latest data in the block is later than the input start time. Data will then be processed normally until at least the input number of seconds of data have been processed.

This limiting process is implemented in Subroutines INIT, which inputs the limit information from file IESPROLIMITS, and RDPREP, which identifies which data blocks to process.

4.4.4 Program APGA Diagnostic Output.

a. Input Data Diagnostics

The shell diagnostic switch in file IESCNTRLFILE controls the diagnostic print for the input and decode routines. There are four levels of print. This diagnostic switch should not be turned on during production runs. At level 4, it will generate nearly one full page of print for each second of data processed. Figure 27 is sample of the level 1 diagnostics.

Level 1 - Subroutine call trace with frame time and I/O record number.

Level 2 - Level 1 plus action control flags and decode flags (NEWSAT, NEWREV,..., IQFRME,...).

Level 4 - Level 2 plus a formatted print of the unpacked and decoded OLS data frame contents.

b. Processing Status Diagnostics.

If the QC module diagnostic print flag in file IESCNTRLFILE is set to a non-zero value, a series of status diagnostics will be printed in the standard print file. A list of these diagnostics is given in Section 4.4.2.2 of Volume 2, Programmer's Guide.

c. EP, RPA, DM, SM, and CKL modules.

These modules of the APGA program can be directed to generate extensive diagnostic output files. These are controlled via the diagnostic output flags loaded in file IESCNTRLFILE by program LDCON02 (see Section 8.4 of Volume 2, Programmer's Guide). In order to be built, however, these files must be assigned to the run prior to the execution of the APGA program. If

they cannot be opened by the program (in Subroutine OPNDOF), the diagnostic output will not be generated. (Note: This is done in order to prevent the generation of these files in a normal production run.) Figure 28 shows examples of the diagnostic output written to the ASCII format output files (IESEPPRT, IESRPAPRT, IESDMPRT, IESSMPRT, and IESCKLPRT), and Figure 29 shows the format of the output to the binary output files (IESEPDIAG, IESRPADIAG, IESDMDIAG, IESSMDIAG, and IESCKLDIAG). If the diagnostic print flags are negative, any requested ASCII format output will be printed to the standard print file as it is generated.

d. EDR ASCII output.

Figure 30 shows an example of an EDR ASCII output. These will be written to file IESEDRPRT if the diagnostic print flag for the EDR output is set to a non-zero value. This value will identify the skip increment between EDRs output to this file, i.e., if the value is 4, every fourth EDR will be printed out to the diagnostic output file.

e. Transfer file ASCII output.

Figure 31 shows an example of a transfer file record ASCII output. These will be written to file IESXFRPRT if the diagnostic print flag for transfer file output is set to a positive, non-zero number. If set to a negative number, these will be written to the standard print file as they are generated.

```
*** INPUT ***
*** CHEKIT ***
*** EXIT CHEKIT *** IFTIME        28804
*** DECODE ***
*** EXIT DECODE IDTIME        28804 ICCNTR          105 IQFRME            1
*** EXIT INPUT *** NFRAME             5 INTIME         28804 THEEND  F RDSTAT
          0
*** PROCES ***
*** EXIT PROCES ***
*** INPUT ***
*** CHEKIT ***
*** EXIT CHEKIT *** IFTIME        28805
*** DECODE ***
*** EXIT DECODE IDTIME        28805 ICCNTR          106 IQFRME            1
*** EXIT INPUT *** NFRAME             6 INTIME         28805 THEEND  F RDSTAT
          0
*** PROCES ***
*** EXIT PROCES ***
*** INPUT ***
*** CHEKIT ***
*** EXIT CHEKIT *** IFTIME        28806
*** DECODE ***
*** EXIT DECODE IDTIME        28806 ICCNTR          107 IQFRME            1
*** EXIT INPUT *** NFRAME             7 INTIME         28806 THEEND  F RDSTAT
          0
*** PROCES ***
*** EXIT PROCES ***
*** INPUT ***
*** CHEKIT ***
*** EXIT CHEKIT *** IFTIME        28807
*** DECODE ***
*** EXIT DECODE IDTIME        28807 ICCNTR          108 IQFRME            1
*** EXIT INPUT *** NFRAME             8 INTIME         28807 THEEND  F RDSTAT
          0
*** PROCES ***
*** EXIT PROCES ***
```

Figure 27 Sample run diagnostic outputs

```
                    --------------------------------------IESRPAPRT.DAT--------------------------------------
*RPA DIAGNOSTIC OUTPUT**RPA DIAGNOSTIC OUTPUT**RPA DIAGNOSTIC OUTPUT**RPA DIAGNOSTIC OUTPUT**RPA DIAGNOSTIC OUTPUT*
 28803 **RPA** BAD PLASMA ENVIRONMENT CALCULATION ***
 28819 **RPA** HELIUM POTENTIAL TEST FAILURE ***
 28835 **RPA** UNABLE TO DETERMINE ION MODE ***
 28847 **RPA** BAD PLASMA ENVIRONMENT CALCULATION ***
 28855 **RPA** UNABLE TO DETERMINE ION MODE ***
 30655 **RPA** BAD PLASMA ENVIRONMENT CALCULATION ***
 30659 **RPA** FRACTIONAL CURRENT FAILURE ***
 30663 **RPA** BAD PLASMA ENVIRONMENT CALCULATION ***
                  -----------------------------------------------------------------------------------------------

                    --------------------------------------IESEPPRT.DAT--------------------------------------

                                             (not currently available)

                  -----------------------------------------------------------------------------------------------

                    --------------------------------------IESDMPRT.DAT--------------------------------------
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
 DM PROCESSING CONSTANTS
 DMAEFF:  4.215E-04   DMK1/DMK2:   0.162  1.364   DMOOFF:   2.560
 DMAG/DMAOFF:   1.027 10.850   DMBG/DMBOFF:   1.027 10.820                                       Header record (solitary)
 ANGMAX:   45.   DMSH/IDMH:   1.0 1   DMSV/IDMV:   1.0 0
 DSMTG/DSMTO:  64.100  1.934
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
 DATA FRAME TIME: 28812 AVERAGE DEN/UH/UV:    5.83E+04   -545.58   -198.57  IDMAR:   3
 LLA/LLB:  2.92  DM MODE:  0  DMEQP:  0.00  DMVO:  2.53 MONDSM: 128  VTOT:   7424.3  IDMUR:  0
  DMOFF:  2.40 2.11 2.38 2.11 2.38 2.11 2.40 2.09 2.39 2.11 2.42 2.11                            Data record (recurrent)
 UH:     -541.6   -541.6   -541.6   -565.6   -541.6   -541.6
 UV:     -192.6   -216.6   -216.6   -192.6   -204.6   -168.5
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
 DATA FRAME TIME: 28813 AVERAGE DEN/UH/UV:    5.28E+04   -543.57   -202.59  IDMAR:   3
 LLA/LLB:  2.88  DM MODE:  0  DMEQP:  0.00  DMVO:  2.53 MONDSM: 128  VTOT:   7424.3  IDMUR:  0
  DMOFF:  2.41 2.11 2.41 2.10 2.37 2.09 2.39 2.13 2.39 2.12 2.38 2.10
 UH:     -541.6   -553.6   -565.6   -517.5   -529.5   -553.6
 UV:     -180.5   -180.5   -228.7   -204.6   -204.6   -216.6
** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **   ** DMPRC **
                  -----------------------------------------------------------------------------------------------
```

Figure 28 Processing modules sample ASCII diagnostic outputs

```
                    ------------------------------------IESCKLPRT.DAT------------------------------------
** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **  ** CKLPRC **
 LOCATION - GEOGRAPHIC: -24.8  148.8  APEX: -36.2  223.2  TIME (APEX): 18.15  TIME (UT SEC): 28835
 MODEL PARAMETERS - A/B:    8.000  1.000  DELTA:    0.0  LEFF:    4.500E+06  P1M:    1.50  KP: 3.0  VD USE:  0
 NUMBER OF DATA POINTS IN: DN -  512   PDS -  316
 FFT PARAMETERS: FL/FU =   0.500   10.0   WINDOW FLAG:  3   DETREND FLAG: 0   SMOOTH FLAG:  3   PDS FLAG: 0
 CALCULATED PARAMETERS - VEFF:  6358.5  RMSDN: 2.6748E+02  DNON(%):     0.806  T1: 1.3115E+15  P1:   1.0000  CKL: 2.9509E+30
 DECIMATED POWER DENSITY SPECTRUM
      FREQUENCY       PSD       FREQUENCY       PSD       FREQUENCY       PSD
         0.1800    4.57E+03        8.5000    7.67E+02   850.0000    4.34E+09
         0.3900    1.91E+03       18.0000    1.78E+09  1800.0000    2.44E+09
         0.8500    2.77E+03       39.0000    3.92E+09     0.0000   -1.00E+37
         1.8000    1.30E+03       85.0000    2.90E+09     0.0000   -1.00E+37
         3.9000    2.32E+03      390.0000    1.19E+09     0.0000   -1.00E+37
                  ------------------------------------------------------------------------------------------

                    ------------------------------------IESSMPRT.DAT------------------------------------
 ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **
 SMAEFF: 2.553E-3
 SMEG: 6.270E-9  SMAG/SMAOFF:   10.06 2.703  SMRFIT: 0.015
 VTABLE
   0.70   0.60   0.50   0.40   0.30
   1.49   1.39   1.29   1.19   1.09
   2.28   2.18   2.08   1.98   1.88
   3.08   2.98   2.88   2.78   2.68                                                         Preamble (first report only)
   3.88   3.78   3.68   3.58   3.48
 ETABLE:    0.26    0.22    0.20    0.16    0.12
   SMFG: 0.6530
 SMFOFF:    11.170  11.090  11.050  11.110  11.050  11.080    0.000    0.000    0.000
 SMFFRQ:      18.0    39.0    85.0   390.0   850.0  1800.0      0.0      0.0      0.0
  SMFBW:      14.0    31.0    64.0   303.0   668.0  1360.0      0.0      0.0      0.0
 NSMFLT:  6
 DATA FRAME TIME: 28800 RELWB1: -.01 IRDELR:  0 IWBR:  1 IVDELR:  0 MONDSM:   0 ISMAR:  0 ISMEF:  0 ISMUR:  0 SMADEN:  -1.00E+37
 SMIDEN:  -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37
          -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37
    SMV:       2.95      2.77      2.90      2.75      2.77      2.80      2.88      2.90      2.80      2.93      2.89      2.89
               2.73      2.87      2.91      2.83      2.92      2.87      2.95      2.91      2.93      2.74      2.85      2.93
 SMFPDS:  -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37  0.00E+00  0.00E+00  0.00E+00
  SMFLT:       2.53      2.93      3.06      3.33      3.96      4.14
 ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **
 ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **
 DATA FRAME TIME: 28801 RELWB1: 3.71 IRDELR:  3 IWBR:  5 IVDELR:  0 MONDSM: 128 ISMAR:  0 ISMEF:  0 ISMUR:  0 SMADEN:  -1.00E+37
 SMIDEN:  -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37
          -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37
    SMV:       2.95      2.79      2.96      2.75      2.80      2.95      2.82      2.92      2.64      2.86      2.78      2.86
               2.78      2.82      2.90      2.75      2.81      2.70      2.85      2.89      2.81      2.75      2.88      2.97
 SMFPDS:  -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37 -1.00E+37  0.00E+00  0.00E+00  0.00E+00
  SMFLT:       2.68      3.20      3.07      3.32      4.00      4.11
 ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **   ** SMPRC **
                  ------------------------------------------------------------------------------------------
```

Figure 28 Processing modules sample ASCII diagnostic outputs (continued)

SMPRC Module

File Name: IESSMDIAG

I/O Record Size: 68 words

Record Format:

word	Variable	Description
1	IDTIME	Data frame time (seconds since midnight)
2	IFLT	Satellite flight ID
3-26	SMV	EI/AMP voltages
27-35	SMFLT	Filter bank voltages
36-59	SMIDEN	Ion densities from SMV (ion/cm^3)
60-68	SMFPDS	PDS estimates from SMFLT ((ion/$cm^3)^2$/Hz)

EPPRC Module

File Name: IESEPDIAG

I/O Record Size: 199 words

Record Format:

word	Variable	Description
1	IEPTME	Sweep center time (seconds since midnight)
2	IFLT	Satellite flight ID
3	VBIAS	V_{bias} voltage
4	MODEP	EP sensor mode (0-6 : A-E)
5	EPDEN	Electron density (el/cm^3)
6	EPTMP	Electron temperature (°K)
7	EPVPOT	Satellite potential (volts)
8-103	VOLT	Sweep voltages
104-199	C	Sweep current (amperes)

Figure 29 Formats of binary diagnostic output files

RPAPRC Module

File Name: IESRPADIAG

I/O Record Size: 201 words

Record Format:

word	Variable	Description
1	IRPTME	Sweep center time (seconds since midnight)
2	IFLT	Satellite flight ID
3	VBIAS	V_{bias} voltage
4	RPODEN	O+ Ion density (ion/cm^3)
5	RPHDEN	Light ion density (ion/cm^3)
6	IRPLIF	Light ion flag
7	RPITMP	Ion temperature (°K)
8	RPAUR	Ram ion drift velocity, u_r (m/s)
9	RPVPOT	Satellite potential (volts)
10-105	VOLT	Sweep voltages
106-201	CURR	Sweep current (amperes)

DMPRC Module

File Name: IESDMDIAG

I/O Record Size: 30 words

Record Format:

word	Variable	Description
1	IDTIME	Data frame time (seconds since midnight)
2	IFLT	Satellite flight ID
3	DMLL	DMLLA/LLB voltage
4	DMIDEN	Ion density derived from DMLL (ion/cm)
5	IDMH	Start location of u_h in DMOFF (0 or 1)
6	IDMV	Start location of u_v in DMOFF (1 or 0)
7-18	DMOFF	DM offset voltages
19-24	DMUH	Horizontal ion drift velocity, u_h (m/s)
25-30	DMUV	Vertical ion drift velocity, u_v (m/s)

Figure 29 Formats of binary diagnostic output files (continued)

CKLPRC Module

File Name: IESCKLDIAG

I/O Record Size: 1840 words

Record Format:

word	Variable	Description
1	ITCKL	Analysis time (seconds since midnight)
2	IFLT	Satellite flight ID
3	APXLAT	Satellite Apex latitude (deg)
4	APXLT	Satellite Apex local time (hours)
5,6	A,B	Model Irregularity axial ratios
7	DELTA	Model delta angle (deg)
8	L	Model effective layer thickness (km)
9	P1M	Model value for p_1
10	VEFF	Effective scan velocity (m/s)
11	ISWCKL	C_kL data source
12	NDN	Number of data points in array DN
13	NPDS	Number of data points in arrays FREQ and PDS
14	T1	T_1 parameter
15	P1	p_1 parameter
16	CKL	C_kL parameter
17	FL	Low-frequency cutoff for log-linear PDS fit
18	FU	High-frequency cutoff for log-linear PDS fit
19-546	DENDTA	Raw density data (528 points) (ion/cm^3)
547-636	FILDTA	SM PDS data (10 9-point sets) ((ion/cm^3)^2/Hz)
637-1148	DN	Detrended density data (NDN points) (ion/cm^3)
1149-1494	FREQ	PDS frequency array (NPDS points) (Hz)
1495-1840	PDS	PDS array (NPDS points) ((ion/cm^3)^2/Hz)

Figure 29 Formats of binary diagnostic output files (continued)

```
**EDR IN LOCATION 3, RECORD 0001, SECTION 0001, SATELLITE F11, DATE: 931025, TIME: 0802
 GLAT    GLON   APXLAT  APXLON   APXLT   ALT
-19.80  147.70  -30.80  221.22   18.04  857.63
-18.66  147.32  -29.58  220.64   18.01  857.32
-17.51  146.94  -28.36  220.07   17.97  857.01
VAPER( 1):      17.44    18.40    17.60    17.12    19.52    17.28    17.60  (VIP =    0.0)
        19.84   17.44    16.80    17.28    17.12    17.44    17.28    17.76
PLASMA DENSITY (SOURCE:  1)
    3.21E+04  3.21E+04  3.20E+04  3.19E+04  3.18E+04  3.19E+04  3.19E+04  3.19E+04  3.19E+04  3.18E+04  3.19E+04  3.18E+04
    3.17E+04  3.17E+04  3.18E+04  3.19E+04  3.19E+04  3.19E+04  3.18E+04  3.17E+04  3.18E+04  3.18E+04  3.18E+04  3.19E+04
    3.19E+04  3.19E+04  3.19E+04  3.19E+04  3.19E+04  3.19E+04  3.20E+04  3.20E+04  3.20E+04  3.19E+04  3.20E+04  3.19E+04
    3.19E+04  3.19E+04  3.20E+04  3.21E+04  3.21E+04  3.20E+04  3.20E+04  3.19E+04  3.19E+04  3.18E+04  3.19E+04  3.20E+04
    3.20E+04  3.20E+04  3.20E+04  3.20E+04  3.22E+04  3.22E+04  3.23E+04  3.23E+04  3.25E+04  3.25E+04  3.25E+04  3.25E+04
HORIZONTAL DRIFT VELOCITY:
    -552.  -560.  -570.  -562.  -562.  -566.  -562.  -566.  -562.  -564.  -548.  -554.  -560.  -552.  -552.
    -548.  -556.  -550.  -550.  -550.  -540.  -546.  -550.  -548.  -552.  -548.  -564.  -552.  -554.  -556.
    -554.  -552.  -546.  -554.  -552.  -558.  -566.  -548.  -550.  -550.  -552.  -558.  -556.  -560.  -560.
    -566.  -566.  -552.  -556.  -556.  -564.  -556.  -558.  -558.  -554.  -558.  -552.  -548.  -550.  -548.
VERTICAL DRIFT VELOCITY:
   -194.6 -190.6 -184.6 -192.6 -190.6 -184.6 -188.6 -178.6 -190.6 -188.6 -180.6 -190.6 -188.6 -192.6 -184.6
   -190.6 -186.6 -192.6 -184.6 -194.6 -194.6 -192.6 -192.6 -188.6 -188.6 -190.6 -184.6 -186.6 -188.6 -188.6
   -182.6 -184.6 -186.6 -188.6 -184.6 -188.6 -186.6 -182.6 -174.5 -176.6 -180.6 -176.6 -178.6 -182.6 -192.6
   -190.6 -180.6 -188.6 -190.6 -196.6 -182.6 -188.6 -178.6 -188.6 -184.6 -190.6 -176.6 -182.6 -192.6 -184.6
CKL ANALYSIS RESULTS (DATA SOURCE:  2)
TIME QUAL  DN/N%      T1        P1       CKL       0.18       0.39       0.85       1.80       3.90       8.50
 05   4     1.00  1.523E+15  1.000  3.426E+30  7.725E+03  2.991E+03  1.971E+03  1.693E+03  2.705E+03  6.016E+02
 15   4     0.94  1.364E+15  1.000  3.068E+30  7.705E+03  2.307E+03  2.336E+03  1.048E+03  1.152E+03  9.961E+02
 25   4     0.91  1.417E+15  1.000  3.188E+30  3.017E+03  1.391E+03  1.716E+03  9.406E+02  1.743E+03  1.298E+03
 35   4     0.99  1.834E+15  1.000  4.127E+30  6.946E+03  2.175E+03  2.344E+03  4.640E+02  1.668E+03  2.968E+03
 45   4     1.08  1.707E+15  1.000  3.840E+30  1.008E+04  1.490E+03  2.013E+03  1.031E+03  1.618E+03  1.900E+03
 55   4     1.06  1.852E+15  1.000  4.165E+30  6.705E+03  2.975E+03  3.616E+03  1.100E+03  1.064E+03  5.641E+02
TIME      18.        39.        85.        390.       850.       1800.       0.         0.         0.
 05    1.975E+09  5.506E+09  3.836E+09  1.512E+09  5.174E+09  2.921E+09 -1.000E+37 -1.000E+37 -1.000E+37
 15    2.484E+09  4.737E+09  3.648E+09  1.293E+09  4.893E+09  2.452E+09 -1.000E+37 -1.000E+37 -1.000E+37
 25    2.020E+09  5.104E+09  3.503E+09  1.350E+09  4.037E+09  2.405E+09 -1.000E+37 -1.000E+37 -1.000E+37
 35    1.954E+09  4.374E+09  3.120E+09  1.355E+09  3.937E+09  2.399E+09 -1.000E+37 -1.000E+37 -1.000E+37
 45    2.801E+09  5.055E+09  3.916E+09  1.548E+09  5.063E+09  2.652E+09 -1.000E+37 -1.000E+37 -1.000E+37
 55    2.749E+09  5.611E+09  4.333E+09  1.521E+09  5.880E+09  2.648E+09 -1.000E+37 -1.000E+37 -1.000E+37
EP SWEEP ANALYSIS DATA(SOURCE:  2)
   TIME     EDEN      TE     VPOT   QUAL
  28922 -1.00E+37***************     0
  28926 -1.00E+37***************     0
  28930 -1.00E+37***************     0
  28934 -1.00E+37***************     0
  28938 -1.00E+37***************     0
  28942 -1.00E+37***************     0
  28946 -1.00E+37***************     0
  28950 -1.00E+37***************     0
  28954 -1.00E+37***************     0
  28958 -1.00E+37***************     0
  28962 -1.00E+37***************     0
  28966 -1.00E+37***************     0
  28970 -1.00E+37***************     0
  28974 -1.00E+37***************     0
RPA SWEEP ANALYSIS DATA (SOURCE:  1)
   TIME    N(O+)      N(H+)  LIF   TI       UR   QUAL
  28926  2.70E+04 -1.00E+37  0 1973.3      0.     1
  28930  1.53E+04  1.33E+04  1 1778.0   -764.     1
  28942  1.76E+04  1.19E+04  1 3762.2  -2418.     1
  28954  2.73E+04 -1.00E+37  0  857.3      0.     1
  28962  2.92E+04 -1.00E+37  0  548.8      0.     1
  28974  1.76E+04  1.21E+04  1 3491.6  -1060.     1
  28978  2.02E+04  8.21E+03  2 1691.1  -1615.     1
DM DENSITY DATA:
    5.44E+04  4.70E+04  5.82E+04  5.30E+04  5.80E+04  5.31E+04  5.04E+04  5.19E+04  5.03E+04  5.18E+04  5.19E+04  5.06E+04
    5.18E+04  5.06E+04  5.16E+04  5.55E+04  5.16E+04  5.53E+04  5.41E+04  4.28E+04  5.43E+04  4.29E+04  4.69E+04  4.70E+04
    4.71E+04  4.69E+04  5.53E+04  4.82E+04  5.53E+04  4.81E+04  5.18E+04  4.29E+04  5.18E+04  4.28E+04  5.19E+04  5.18E+04
    5.41E+04  4.94E+04  5.40E+04  4.94E+04  5.40E+04  4.94E+04  5.15E+04  4.93E+04  5.19E+04  4.93E+04  5.14E+04  4.93E+04
    5.18E+04  4.91E+04  5.54E+04  5.41E+04  5.54E+04  5.44E+04  5.68E+04  4.59E+04  5.66E+04  4.60E+04  5.30E+04  5.42E+04
MISCELLANEOUS INFORMATION
   ELECT TEMP:   0.00,  ADC TEMP:  34.25,  DSM TEMP: -45.11,  DM VOLTAGE:   2.53,  DM MODE:  0,  EP MODE:  2
```

Figure 30 Sample EDR ASCII diagnostic output

4.4.5 Initializing Fortran 77 Direct Access Files.

The four direct-access files used by the APGA system,

IESEDRFILE, IESSTATFILE, IESAGDBXFR1, IESAGDBXFR2

are opened and initialized with a (now meaningless) header record by the APGA program. These files are direct access due to the complicated processing and multi-access requirements of the GWC Sperry database system. The header record is also a remnant of GWC Sperry processing.

4.5 Listings.

The complete set of program listing for the SSIES Flight Data Processing System (programs APGA, LDCON02, and BNBA) is maintained by AFSFC.

IESXFERPRT.DAT

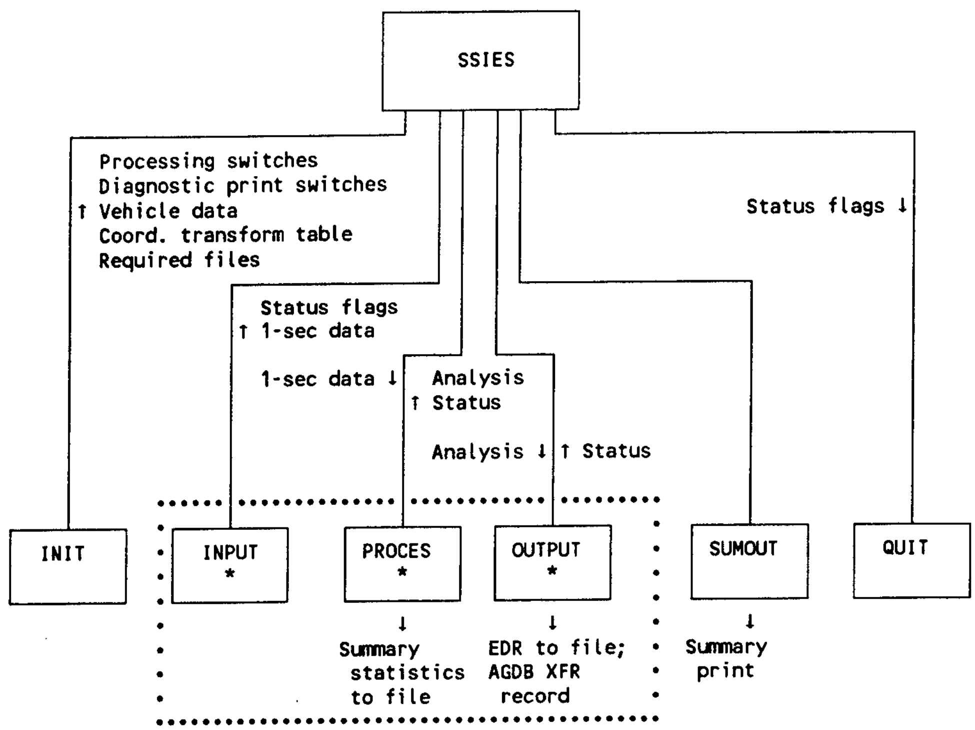
Date (YYMMDD) | UT (HHMM) | Latitude | Longitude | Ne | Ne source | Ion ratio | Te | Ti

```
XFER: 931025 0806  -5.8 144.3 6.21E+4 1  0.166****** 2024.
         2    4.80E+30    1.00  7.20E+30   1.00  8.64E+30   1.00  9.68E+30   1.00  1.29E+31   1.00  1.32E+31   1.00
XFER: 931025 0807  -2.3 143.5 7.02E+4 1  0.607****** 2216.
         2    9.50E+30    1.00  9.46E+30   1.00  1.07E+31   1.00  1.40E+31   1.00  2.12E+31   1.00  2.54E+31   1.00
XFER: 931025 0808   1.4 142.7 6.63E+4 1  0.137****** 1773.
         2    2.36E+31    1.00  2.28E+31   1.00  3.00E+31   1.00  3.60E+31   1.00  3.85E+31   1.00  3.62E+31   1.00
```

CKL source | CKL ... P1 (00-10 sec) | CKL ... P1 (10-20 sec) | CKL ... P1 (20-30 sec) | CKL ... P1 (30-40 sec) | CKL ... P1 (40-50 sec) | CKL ... P1 (50-60 sec)

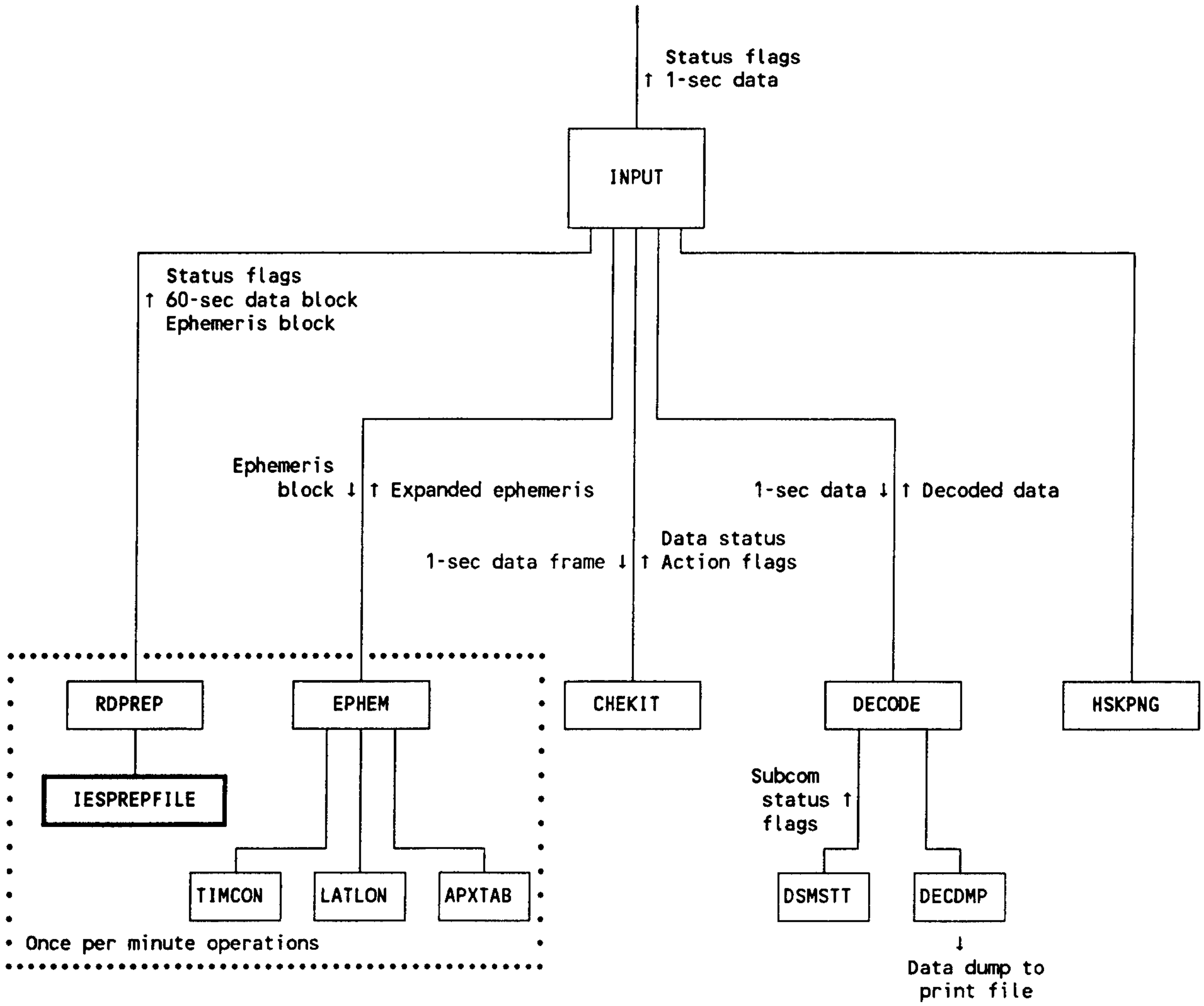
Figure 31 Sample AGDB transfer record ASCII diagnostic output

APPENDIX A: Program APGA Structure Charts

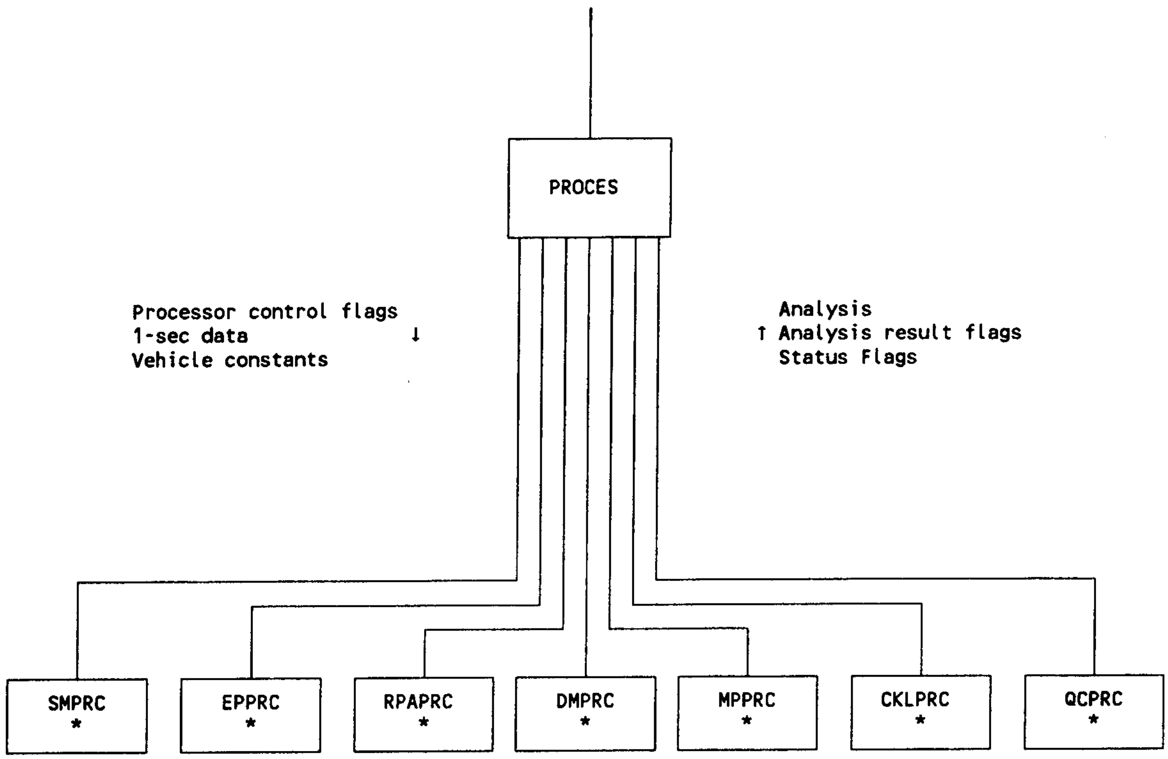


* = See Separate Structure Chart

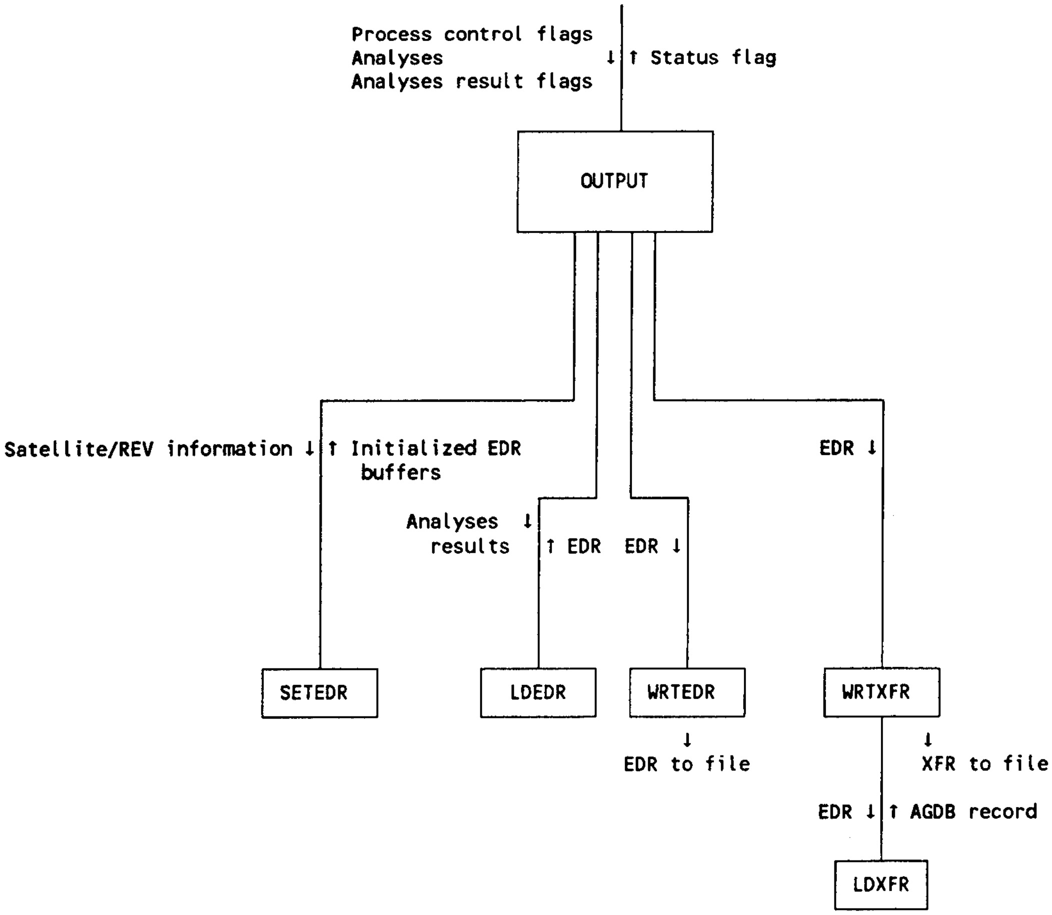
Dotted box indicates recurrent section for processing each second of data.



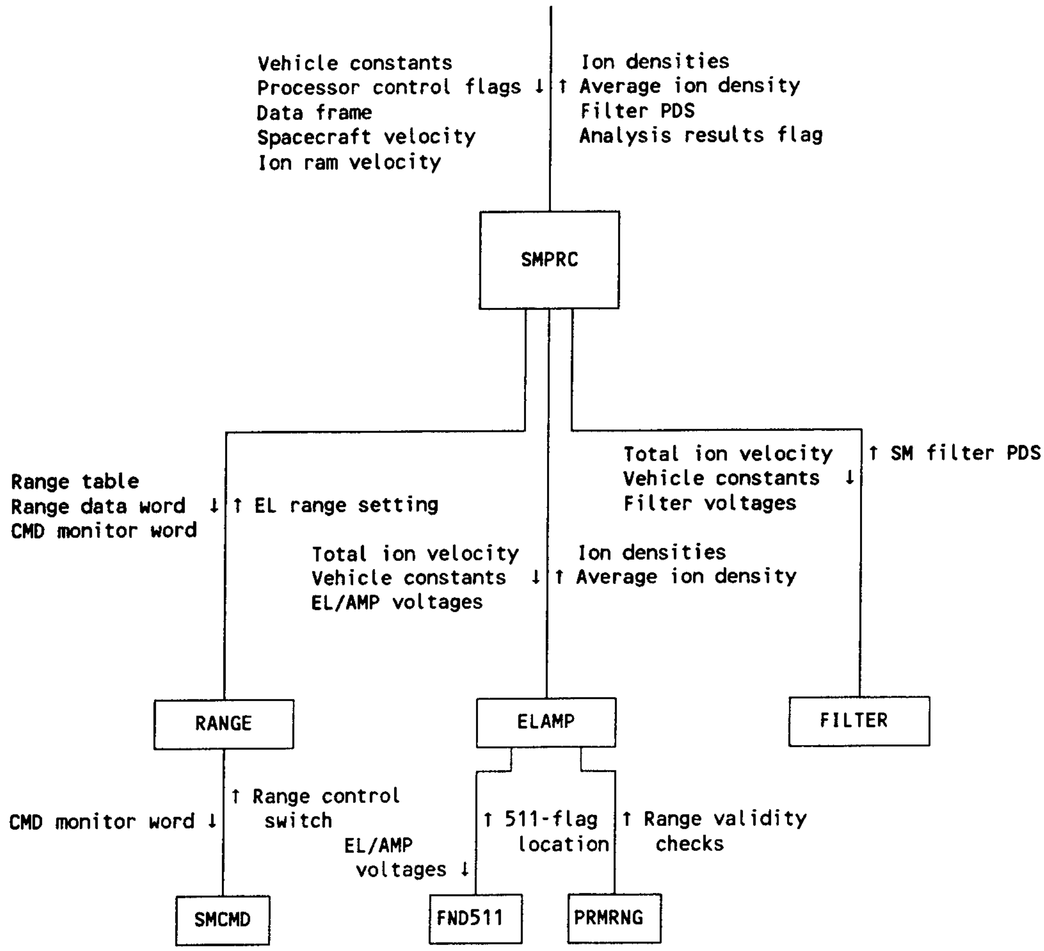
Status flags
↑ 1-sec data
INPUT
Status flags
↑ 60-sec data block
Ephemeris block
Ephemeris
block ↓
↑ Expanded ephemeris
1-sec data frame ↓
Data status
↑ Action flags
1-sec data ↓
↑ Decoded data
RDPREP
IESPREPFILE
EPHEM
TIMCON
LATLON
APXTAB
Once per minute operations
CHEKIT
DECODE
Subcom
status ↑
flags
DSMSTT
DECDMP
↓
Data dump to
print file
HSKPNG



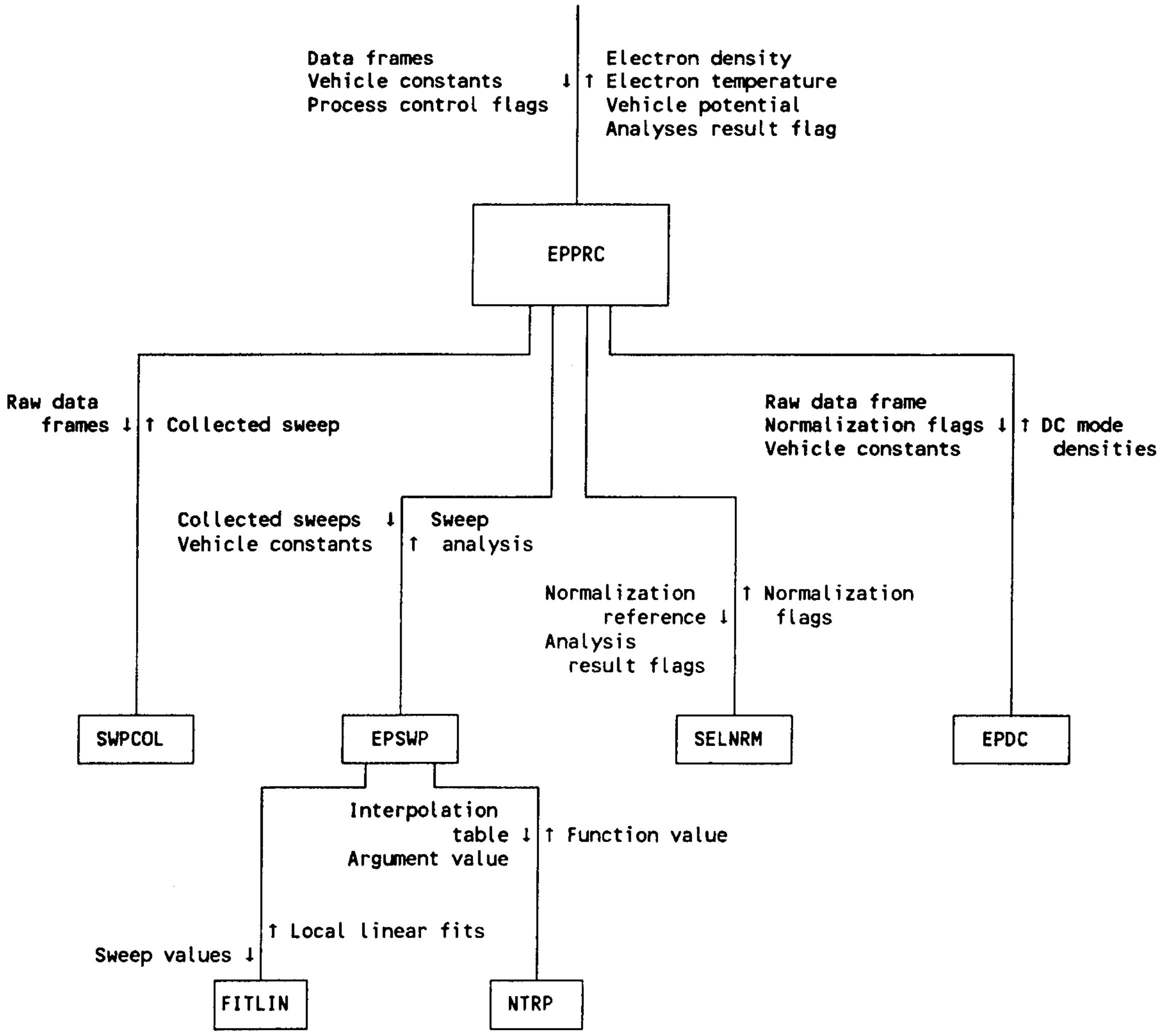
* = See Separate Structure Chart



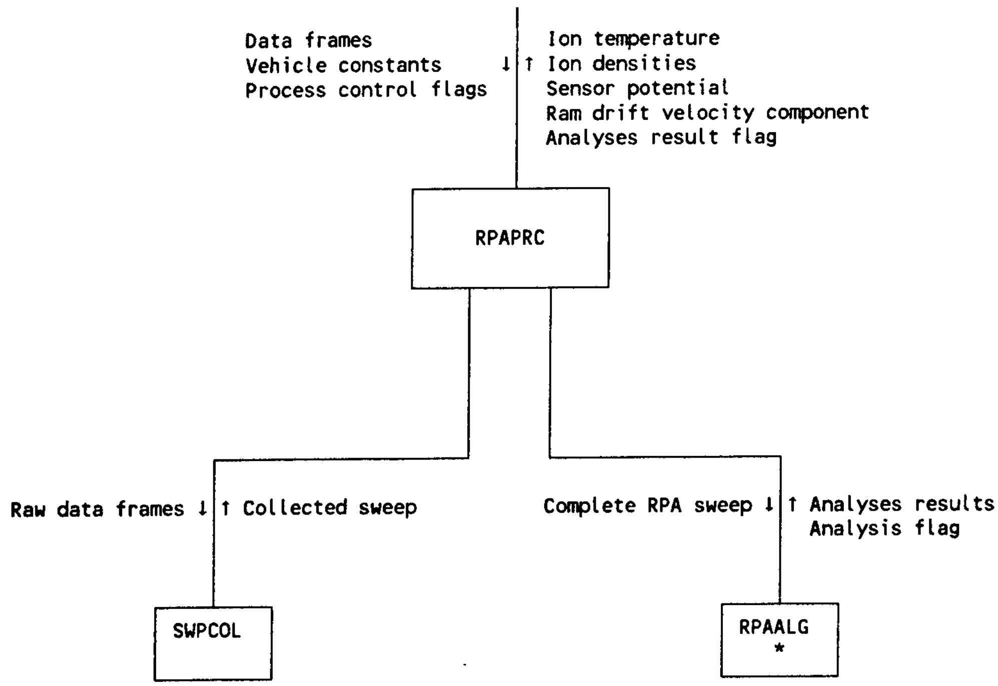
Some second-level subroutines are not shown.



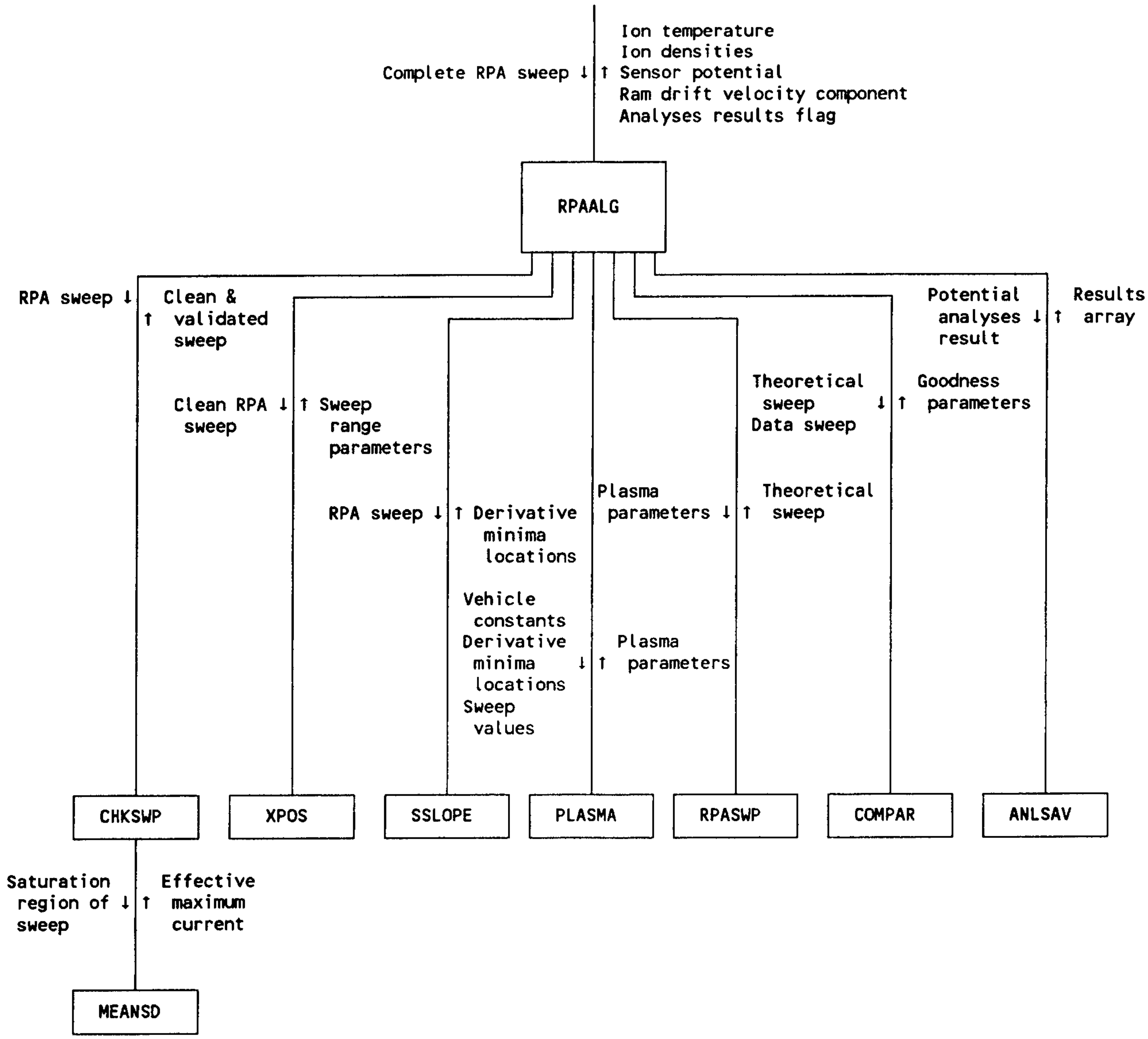
Some utility and second-level subroutines are not shown.



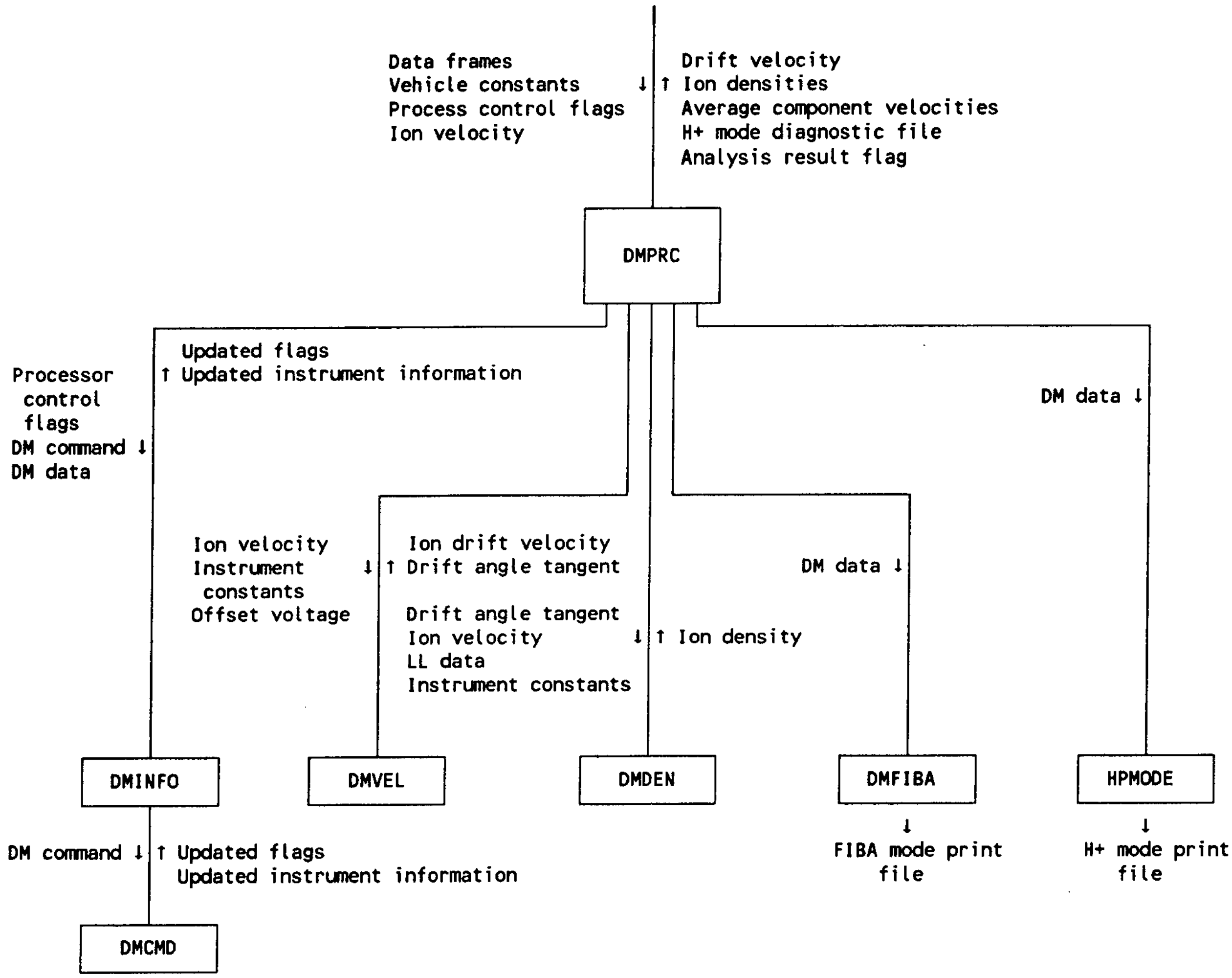
Some utility and second-level subroutines are not shown.



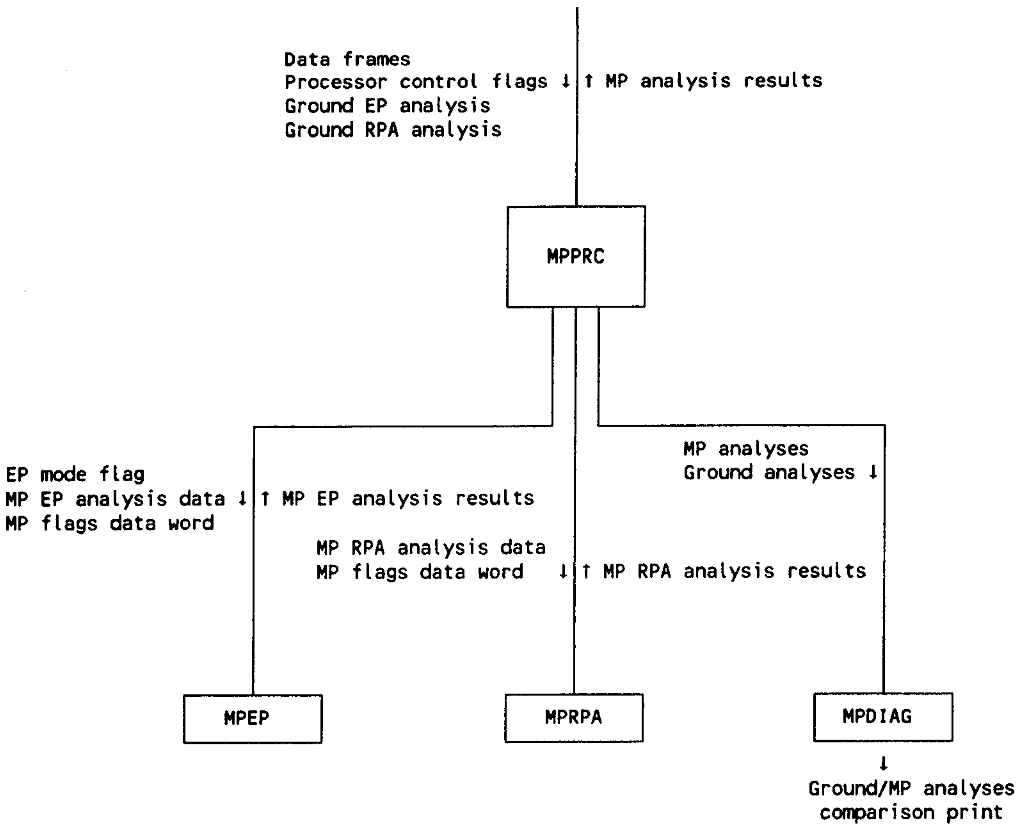
* = See Separate Structure Chart



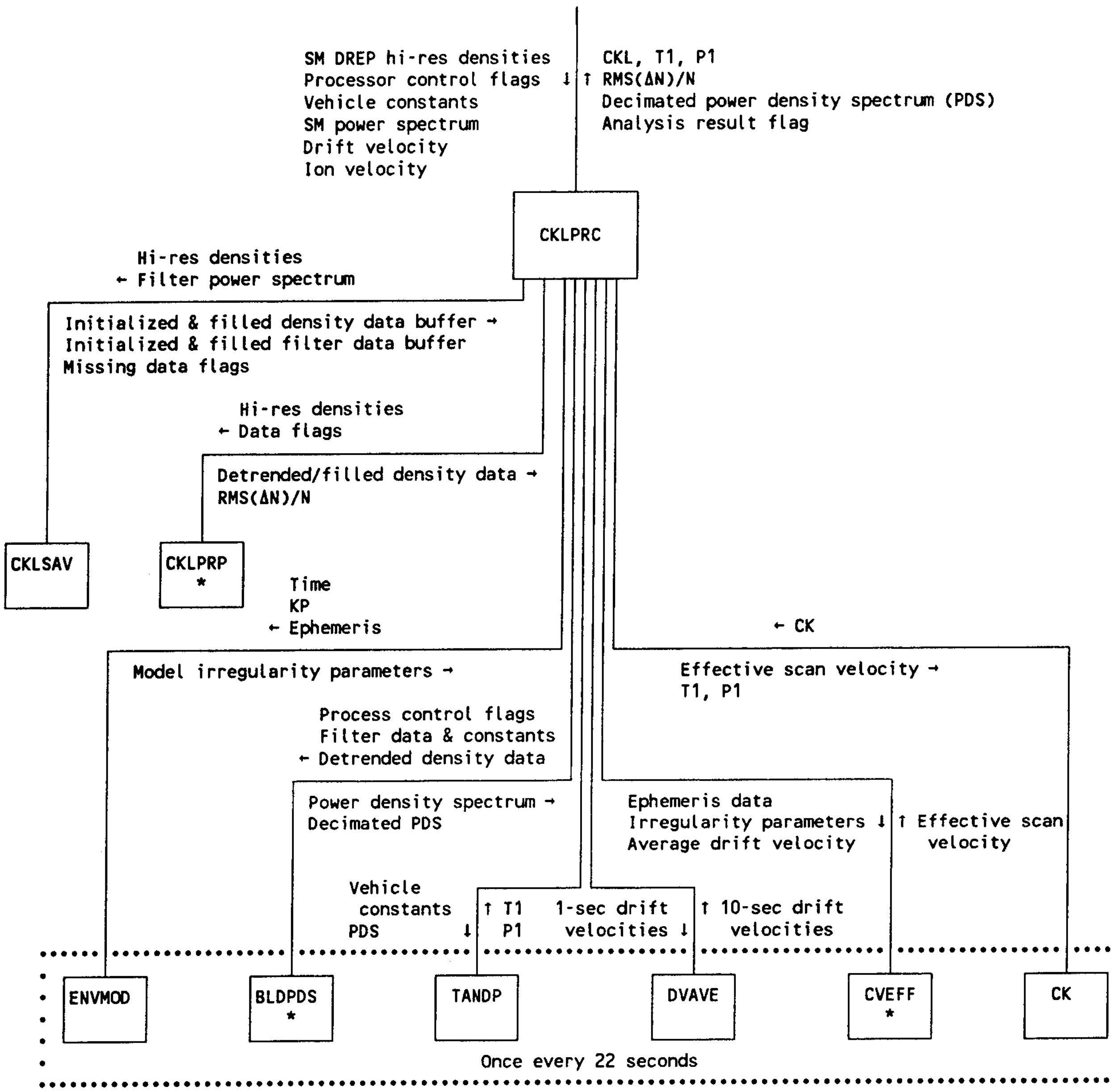
Some utility and second-level subroutines are not shown.



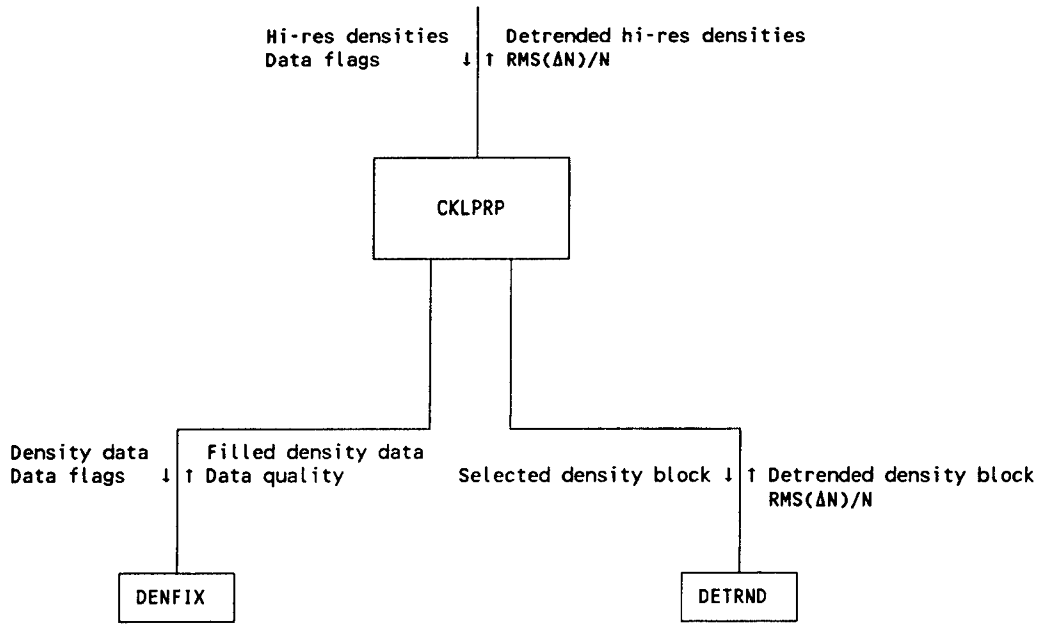
Some utility and second-level subroutines are not shown.



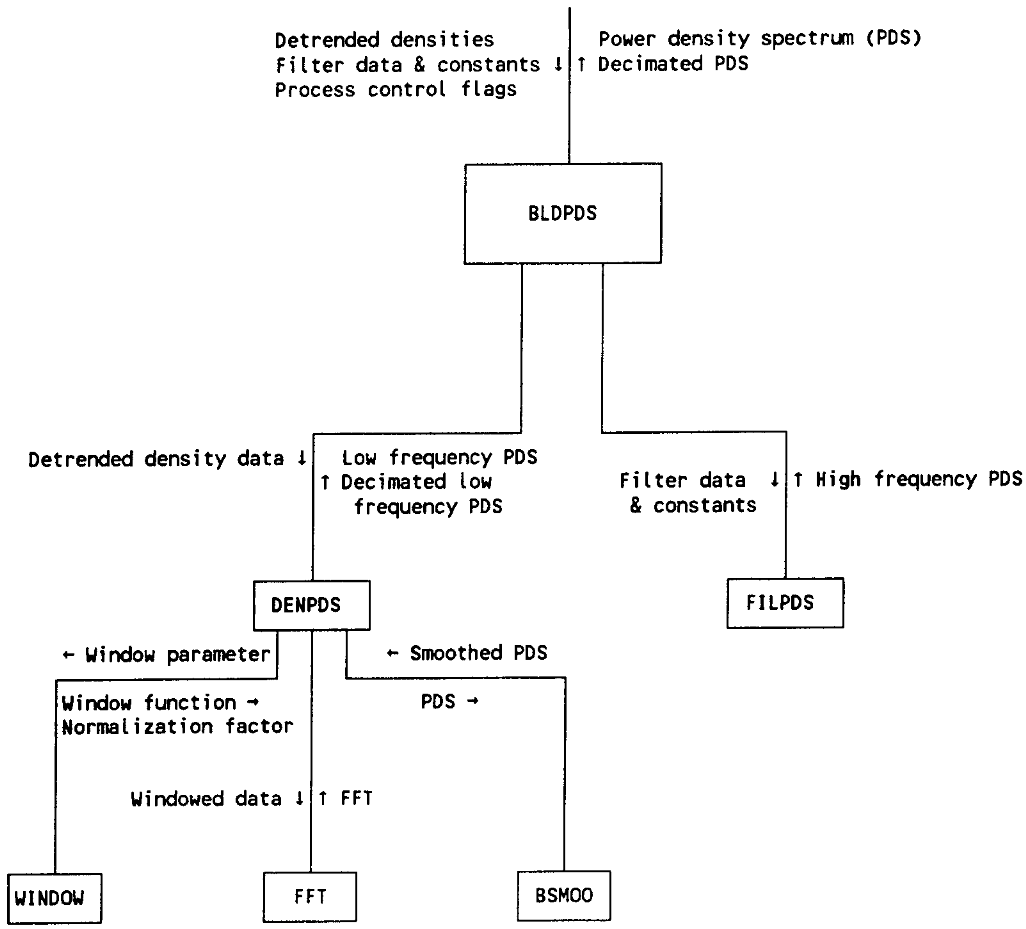
Some utility and second-level subroutines are not shown.



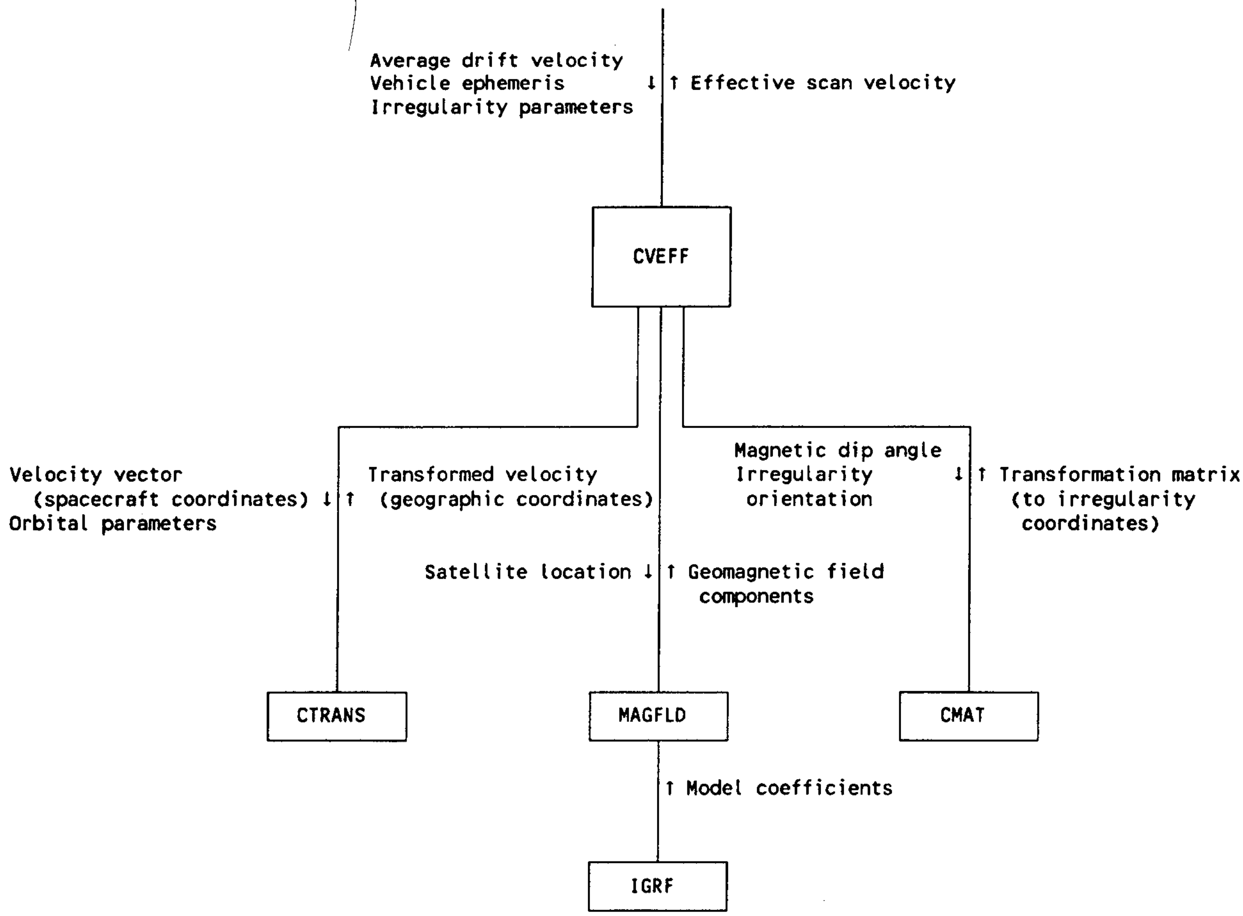
* = See Separate Structure Chart
Some utility and second-level subroutines are not shown.



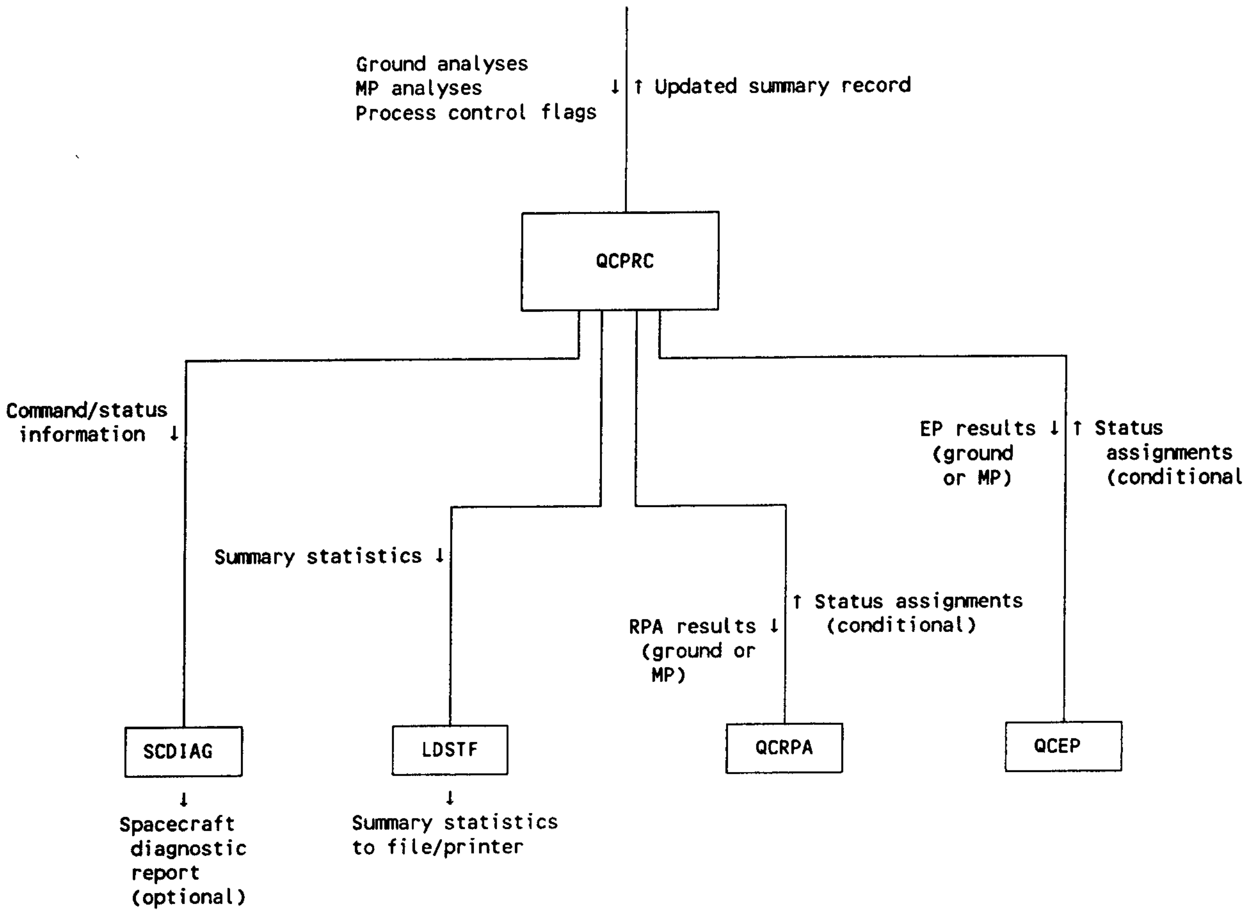
Some utility subroutines are not shown.



Some utility and second-level subroutines are not shown.



Average drift velocity
Vehicle ephemeris
Irregularity parameters
Effective scan velocity
CVEFF
Velocity vector
(spacecraft coordinates)
Orbital parameters
Transformed velocity
(geographic coordinates)
Magnetic dip angle
Irregularity
orientation
Transformation matrix
(to irregularity
coordinates)
Satellite location
Geomagnetic field
components
CTRANS
MAGFLD
CMAT
Model coefficients
IGRF


Ground analyses
MP analyses
Process control flags
Updated summary record
QCPRC
Command/status information
Summary statistics
EP results (ground or MP)
Status assignments (conditional)
RPA results (ground or MP)
Status assignments (conditional)
SCDIAG
LDSTF
QCRPA
QCEP
Spacecraft diagnostic report (optional)
Summary statistics to file/printer

Process control flags
Diagnostic print flags
Irregularity model information
Data frame conversion factors
Process control information
Instrument constants

↓

LDCON02

↓ All input information to
binary record in IESCNTRLFILE

IESCNTRLFILE

APPENDIX B: Description of Data Analysis Algorithms

B.1 Electron Probe (EP) Processing Algorithms.

a. Sensor Description.

The electron sensor is a spherical Langmuir probe which measures the current to a collector exposed to the environment as a function of the electrostatic potential on the collector. The sensor has five operational modes. Figure B1 depicts the electron sensor modes. The modes can be characterized as either sweep (modes A, B, BS, and E) or non-sweep (modes C, D, and DS).

During all modes there is a periodic set of calibration sweeps to enable the sensor to determine a proper bias voltage. These sweeps cover a broader voltage range as a coarser voltage step than the normal mode sweeps. This helps assure that the instrument can locate a suitable bias voltage, but results in a less accurate determination of the desired environmental parameters when processed as a normal sweep.

The sweep modes allow calculation of the electron temperature, the sensor potential, the absolute electron temperature, and the absolute electron density. These values can be calculated at 4 second intervals in modes A, B, and BS, or at 2 second intervals in mode E.

In the non-sweep modes, the instrument measures the change in density as function of time. A current measurement is accomplished 24 times each second. To be useful, a density value must be periodically available so that the current can be scaled to provide the corresponding density values at the higher sampling rate. To accomplish this, even in these "non-sweep" modes, a sweep is performed at 32 second intervals to provide an absolute density value to which the intervening current measurements can be normalized.

A more complete description of the electron sensor and its operational modes is provided in Volume 1, Technical Description.

b. Sensor Output.

The EP sensor generates 30 data words each second, 6 voltages and 24 currents. It takes four seconds to accumulate a complete normal sweep, three seconds to accumulate a calibration sweep and 2 seconds to accumulate a sweep in the short sweep mode (mode E). In the DC mode, successive currents represent a change in the electron density. These currents can be normalized to produce absolute density values. Normalization can be to the periodic sweep in the EP data occurring every 32 seconds in the DC mode, or to any other source of density data available such as the RPA or the SM.

Because of limitations on the telemetry for vehicles beginning with F11 (SSIES-2 and SSIES-3 instruments), the EP sensor voltages are not included in the data, but are instead provided within the processing program (Subroutine VSWEEP). Furthermore, the EP current values may also be excluded from the telemetry, in favor of the RPA currents, based on command settings for the vehicle.

c. Data Extraction Algorithm.

(1) Sweep processing.

(a) Collect a complete sweep, ensuring that each second of data is part of the same sweep.

(b) When a sweep is fully collected, verify and clean up the sweep. The sweeps are all

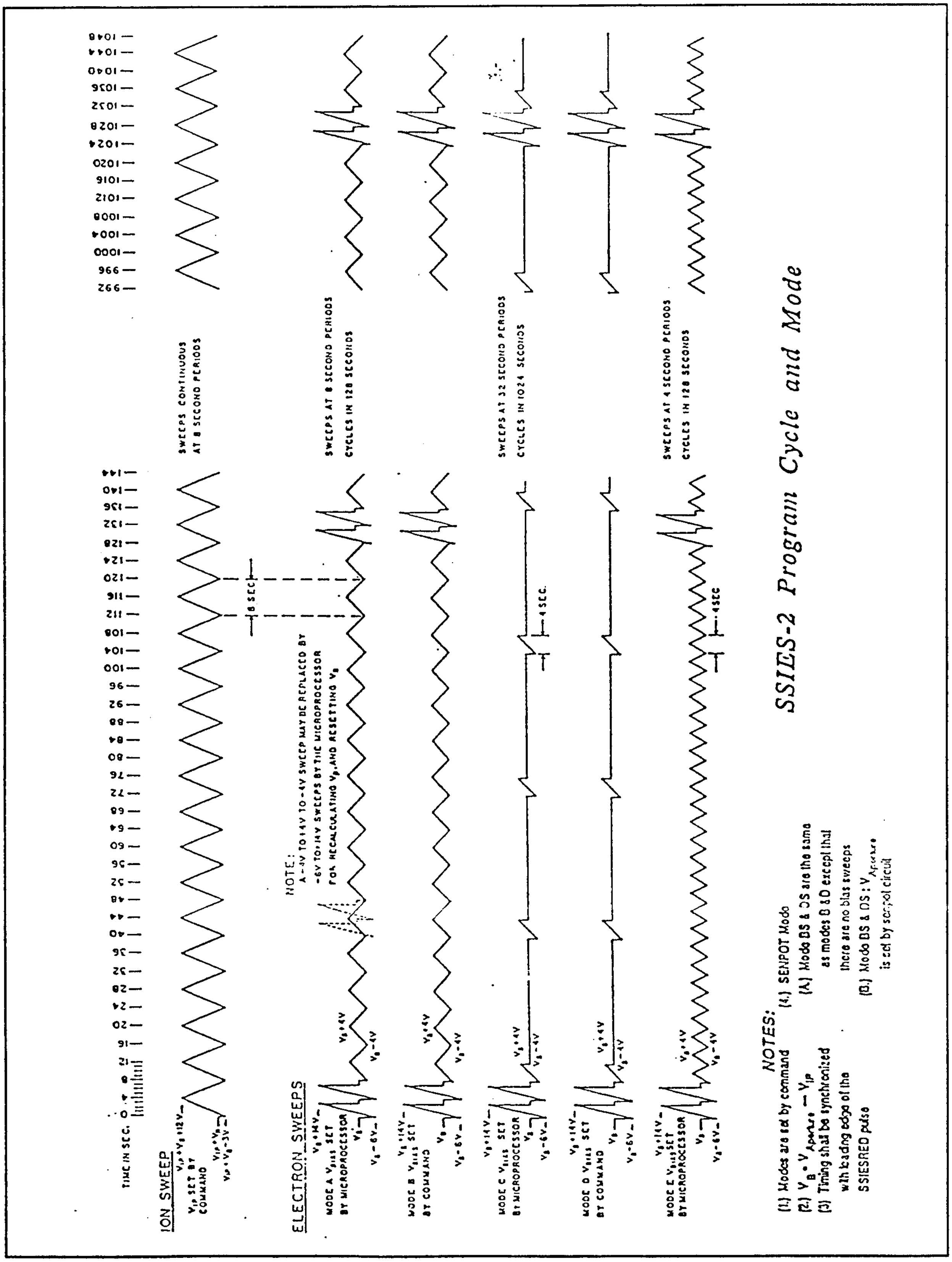


Figure B1 SSIES Electron Sensor modes

processed scanning from the high current (high voltage) end to the low current (low voltage) end. Clean sweeps will be monotonic. However, since the environment is not stable during the entire sweep, and there is round-off error in encoding the data into the telemetry stream, we must allow for current variations in both directions within the expected limits of geophysical and telemetric noise. Volume 1, Technical Description, Figure 6.4.1 depicts a typical EP sweep structure.

(c) Fit a straight line to the start of the acceleration region of the sweep.

(d) Move along the remainder of the sweep, doing a linear fit to locate the maximum slope of the sweep. This point should be the point where the slope first reaches maximum. This indicates the start of the retarding region of the sweep.

(e) Determine the potential at this point by finding the intersection of the lines determined in (c) and (d).

(f) If desired, a quadratic correction can be applied to the determined potential. The intersection of the two lines is systematically offset a slight amount from the actual desired potential.

(g) Calculate the electron temperature from the slope of the line fitted to the retarding region:

$$T_e = 5040 / S$$

where: T_e = electron temperature,
S = slope of fitted line at start of retarding region.

(h) Calculate the electron density from the current at that point:

$$N_e = \frac{2\sqrt{\pi}\, I}{Ae\alpha a}$$

where: I = current at point of maximum slope (e.g., sensor potential),
A = sensor area,
α = sensor transparency,
a = most probable electron speed,
e = electron charge.

(2) DC mode processing.

In DC mode, we directly calculate the electron density by normalizing to the current at the sensor potential during the last EP sweep. The density used for this normalization can be from any density source available. Selection is based upon the availability of a density measurement at the time the normalizing current is measured. A prioritized list is used in the software. Any of the sensors can be disabled/enabled for this purpose by setting a flag in the file IESCNTRLFILE. The selection order is:

(a) Scintillation Meter (SM)
(b) Retarding Potential Analyzer (RPA)
(c) Drift Meter (DM)
(d) Electron Probe (EP)

B.2 Retarding Potential Analyzer (RPA) Processing Algorithms.

a. Sensor Description.

The RPA, as shown in Volume 1, Technical Description, Figure 3.4.2, is a planar ion collector that allows ions to freely enter the aperture. The ground plane around the sensor ensures that the surfaces near the sensor are electrostatically uniform, allowing the entering ion trajectories to be represented by straight lines. Due to exposed contacts on the solar panel, the satellite potential could reach a relatively large negative potential. To alleviate this problem, the ground plane is isolated from the sensor and the electronics by a large resistance. The ground plane floats at some potential with respect to the plasma. This potential is measured by the sensor electronics, which then apply a varying potential to the ground plane to keep the ground plane near the potential of the undisturbed plasma. A detailed description of the sensor can be found in Volume 1, Technical Description.

Internally, the retarding grids are stepped through a series of applied voltages. A single Maxwellian temperature is assumed for all ion species. When the potential barrier is equal to the energy gained by an ion species from the satellite potential and satellite velocity, half the ions of that species will not penetrate the retarding grids. For a spatially uniform plasma, the current collected is constant until the potential applied begins to reject the lightest ion species present in a measurable quantity. Then the current decreases monotonically and the magnitude of the rate of change in current with respect to the change in sweep voltage increases. This rate of change reaches a maximum at the center of the Maxwellian temperature distribution. As the applied voltage further increases, the current becomes constant until the next heavier ion species begins to be rejected. Figure B2 represents a theoretical response for the RPA. If, due to the spacecraft velocity and ion drift velocities, there are species which have their rejection potential near the same voltage, it will be difficult to analyze the RPA data. This happens in the ionosphere when both H+ and He+ are present in significant amounts.

b. Sensor Output.

The RPA sensor generates 30 data words each second, 6 voltages and 24 currents. It takes four seconds to send a complete sweep.

Because of limitations on the telemetry for vehicles beginning with F11 (SSIES-2 and SSIES-3 instruments), the RPA sensor voltages are not included in the data, but are instead provided within the processing program (Subroutine VSWEEP). Furthermore, the RPA current values may also be excluded from the telemetry, in favor of the EP currents, based on command settings for the vehicle.

During the plasma chamber testing of an SSIES instrument, an extra induced current was detected. This displacement current is proportional to the rate of change of the sweep voltage and to the capacitance between the collector and ground. This displacement current is subtracted from the measured current on the upsweep and added to the measured current on the downsweep. Actual values of this displacement current must be determined from flight data.

The SSIES data processor and the data simulator used to test the processor provided for a displacement current. An estimated value for this displacement current was used. The displacement current produced a noticeable effect on the algorithm analysis results at low modelled ion densities.

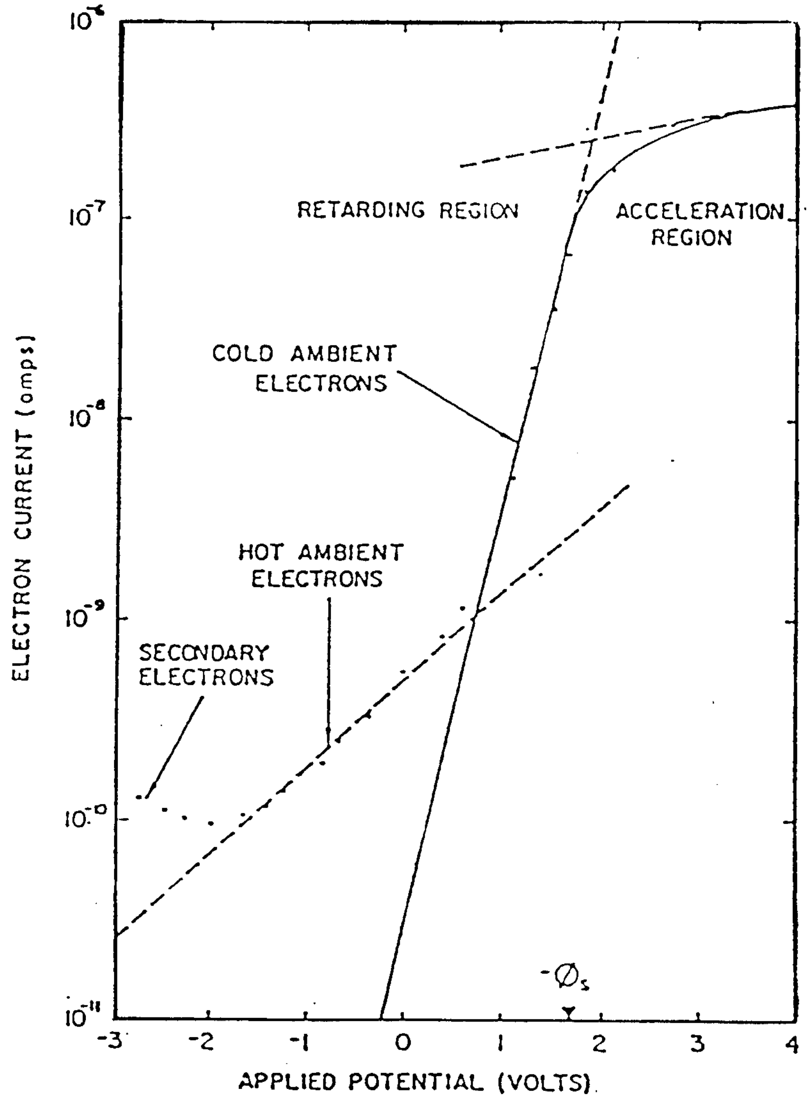


Figure B2 Theoretical instrument response to a 2-ion constituent plasma

c. Data Extraction Algorithms.

(1) Collect a complete sweep, ensuring that each second of data is a part of the same sweep.

(2) When a sweep is fully collected, verify and clean up the sweep. The sweeps are all processed scanning from the high current (low voltage) end to the low current (high voltage) end of the sweep. Clean sweeps will be monotonically increasing in voltage, and, ideally, should be monotonically decreasing in current. However, since the environment is not stable during the entire sweep period, and there is round-off error in encoding the data into the telemetry stream, we must allow for current variations in both directions within reasonably expected limits of geophysical and telemetry noise.

(3) Working from the start (high current end) of the sweep and disregarding any initial saturated currents, fit a straight line to 3 successive data points to determine the slope. This fitting process is repeated for each data point in the sweep to locate positions where the slope reaches a local relative minima. If more than two minima are found, increase the number of points used in the fitting process and scan the sweep again. When we have smoothed the sweep through this process until there are at most two local minima, we can calculate the required environmental parameters.

(4) If there are two local minima, assume that there are both heavy ions (Oxygen) and light ions (Hydrogen and/or Helium) present. The following quantities are used in the analysis

ϕ_i = potential at i^{th} local minima,

I_o = maximum current,

I_i = current at i^{th} local minima,

m_i = mass of i^{th} constituent,

a_i = most probable speed of i^{th} constituent,

S_i = slope at i^{th} local minima,

V_s = satellite velocity,

V_t = total relative velocity between ions and satellite,

A = sensor aperture area,

α = sensor transparency,

z = ionization level number,

e = electron charge,

k = Boltzmann's constant.

(a) Calculate the sensor potential.

$$\phi_s = \frac{\phi_2 m_1 - \phi_1 m_2}{m_2 - m_1}$$

(b) Calculate the ram direction drift velocity.

$$U_r = \left[\frac{2e}{m_2}(\phi_2 + \phi_s)\right]^{1/2} - V_s = V_t - V_s$$

(c) Calculate the heavy ion [O+] density and temperature.

$$N\,[O+] = \frac{\pi m_2}{Ae^2 d} \mid S_2 \mid \left\{ -\frac{V_t}{2} + \left[\frac{V_t^2}{4} + \frac{2\,e\,I\,z}{\pi m_2 \mid S_2 \mid}\right]^{1/2} \right\}$$

$$T\,[O+] = \frac{(Ae\alpha N[O+])^2}{2\pi k m_2 S_2^{\,2}}$$

(d) Calculate the light ion density.

$$N = \frac{2\left(\dfrac{I_o}{Ae\alpha V_t} - N[O+]\right)}{\left[1 + ERF\,\dfrac{V_t}{a_1} + \dfrac{a_1}{\sqrt{\pi}V_t}\,EXP\left(-\dfrac{V_t^2}{a_1^{\,2}}\right)\right]}$$

(5) If only one local minimum is found, assume that only one ion species is present (Oxygen). With only one local minimum in the first derivative of the sweep, we cannot unambiguously calculate all parameters of interest. Therefore, we must make some assumptions.

(a) For geomagnetic mid-latitudes, assume that the ram direction drift velocity is zero and calculate the sensor potential.

$$\phi_s = \frac{V_s^2 m_2}{2e} - \phi_2$$

(b) For geomagnetic polar latitudes, use an estimated sensor potential (try to use the EP sensor analysis results) and calculate the ram direction drift velocity as in the two-species case above.

B.3 Drift Meter Processing Algorithms.

a. Sensor Description.

The ion Drift Meter (DM) is a planar Faraday cup sensor located on the ion-sensor ground plane with the ion RPA and ion Scintillation Meter sensors. The sensor is oriented with its normal parallel to the spacecraft direction-of-travel ($+y_s$). The sensor has a retarding-potential grid which can be set to fixed positive voltages (with respect to the ion-sensor ground plane) to retard light ions (such as H+), or can be driven by a varying potential periodically to retard the light ions. This last mode of operations is called H+ mode, while the other modes in which the grid is set at a fixed potential are called normal modes. An alternative mode, unique to SSIES2, is the DM FIBA (Filter Bank) mode, which is similar to the normal mode, but also generates high-frequency filter outputs. The collector plate at the base of the DM sensor is divided into four electrically isolated segments as shown in Volume 1, Technical Description, Figure 4.1. A detailed description of this sensor can also be found in Volume 1.

b. Sensor Output.

In all operating modes, the DM sensor generates 13 data words every second, 12 offset voltages and 1 voltage from either the LLA (LL for Log Level) or LLB amplifier. In FIBA mode, the 6 DM FIBA data words, occurring every other second, contain additional data for the filter band values. In normal mode, the 12 offset voltages can be used to calculate the cross-track ion drift velocities, u_h and u_v, and the LLA/LLB voltages can be used to calculate the total ion density (if the retarding grid is set to zero potential). The offset voltages alternate between horizontal and vertical measurements. In H+ mode, the first two offset voltages can be used to calculate u_h and u_v, but the remainder of the data, including the LLA/LLB measurements, are not usable until algorithms are developed for processing it. The only other H+ data currently used by the APGA program is the LLA data, which is used in H+ mode as a cycle counter. The DM sensor operates in a four-second cycle while in H+ mode. There is no data word available for a cycle counter, so the LLA output voltage, which is reported in every data-frame cycle 1, is set to 5.11 volts at the start of each four second cycle. There is also currently no algorithm developed for processing the DM FIBA values.

c. Data Extraction Algorithms.

The interpretation of the DM data from the Common Format data frame is complicated by H+ mode operations and the fact that the orientation of the DM sensor in the spacecraft coordinate system determines:

1. whether horizontal or vertical offset voltages are reported first or second in the data frame DM words, and
2. the sign of the velocities derived from the offset voltages.

The operating mode of the DM sensor and the potential on the retarding grid can be determined from the DSM command monitor which is reported in Common Format data frame word 3/cycle 2 (Volume 1, Technical Description, Tables 10.1.3, and 10.2.3 list the DSM commands and the corresponding sensor statuses).

In normal mode operations, the only "trick" to extracting the data is in determining if the first offset voltage reported in each data frame is a horizontal or vertical measurement. This is fixed by the orientation of the sensor on the spacecraft. Two location parameters stored in file IESCNTRLFILE are used to specify where in the data frame DM words the horizontal and vertical offset voltages are located. If horizontal offsets are reported first, the horizontal location

constant should be 0 and the vertical location constant should be 1. If vertical offsets are reported first, these values should be reversed. The orientation of the DM sensor should be provided by USAF Space Division. If there is any doubt of the orientation, the relative magnitudes of the two velocities calculated should be compared. At almost all times, the magnitude of the horizontal drift velocity should be appreciably larger than the magnitude of the vertical drift velocity. If this is not the case in the calculated velocities, the orientation is probably not as advertised, and the values of the location constants should be reversed from their current setting.

The "tricks" to H+ mode operations are twofold. First, care must be taken to determine when the DM sensor is in H+ mode, as the data will be quite different. This is tracked by monitoring the DM command monitor word, the DSM subcom status words, and the values reported in the LLA data word. If either the command monitor or the subcom status indicate that the sensor has been commanded into H+ mode or a 5.11 volt reading is reported in the LLA data word, the DMPRC module assumes that the sensor is either in or about to be in H+ mode and enters a wait status. This wait status will continue until the other H+ mode indicator (command monitor or 5.11 volt LLA reading) is encountered or the subcom cycle restarts. At that time, the sensor is assumed to be in H+ mode (if this is consistent with the subcom-designated mode). The same procedure is followed in going out of H+ mode. When in H+ mode, only the first two offset voltage values from each data frame are processed. These data are a horizontal/vertical offset measurement pair, **but they are in the reverse order in H+ mode cycles 3 and 4 from that reported in normal mode (or in H+ mode cycles 1 and 2).** This is the second "trick" to H+ mode processing.

In DM FIBA mode, the six filter band values are acquired from the odd cycles of the Common Format.

d. Data Reduction Algorithms.

1. Normal Mode.

The angle, α, between the normal to the DM sensor and the horizontal or vertical projection of the velocity vector of the ions can be calculated from the offset voltages from:

$$\tan \alpha = C_d \, (V_{tm} - V_0^d)$$

where:
- C_d is a sensor constant from file IESCNTRLFILE,
- V_{tm} is the offset voltage extracted from the data frame
- V_o^d is the zero voltage for the offset measurements, either from file IESCNTRLFILE or from the value reported in word 15 of the Common Format.

The ion drift velocities can then be calculated from:

$$u_h = S_h \, (v_s + u_r) \tan \alpha_h$$

$$u_v = S_v \, (v_s + u_r) \tan \alpha_v$$

where:
- S_h, S_v are the sign constants (± 1.0) from file IESCNTRLFILE,
- v_s is the spacecraft orbital velocity,
- u_r is the ion ram velocity (from the RPA analysis, if available).

The two sign constants are set up in conjunction with the horizontal/vertical location

constants described above so that the two velocities will be correctly oriented in the spacecraft coordinate system. This calculation is implemented in function DMVEL.

If the retarding grid potential is zero (which can be determined from the DM command monitor, word 3/cycle 2 in the Common Format), the total ion density can be estimated from the LLA/LLB voltages from:

$$N_i = \frac{2I}{eA_d\,(v_s + u_r\,)}$$

where:
- e is the electron charge,
- A_d is the effective area of the DM sensor from file IESCNTRLFILE,
- 2I is the total ion current calculated from:

$$I = \frac{10^{LLA}}{1 + \kappa \tan \alpha} = \frac{10^{LLB}}{1 - \kappa \tan \alpha}$$

where:
- κ is a sensor constant from file IESCNTRLFILE,
- LLA, LLB are the LLA/LLB log currents.

Tan α values are those from the offset measurement reported in the data frame just prior to the LLA/LLB data word.

The LLA/LLB log currents are calculated from the LLA/LLB voltages reported in the data frame from:

$$LLA = G_1 V_{tm} + G_0^1$$

$$LLA = G_2 V_{tm} + G_0^2$$

where the four G values are from file IESCNTRLFILE. This calculation is implemented in function DMDEN.

[Note: See the data reduction description for this sensor in the SSIES Project Notebook. This description was provided by the DM instrument builders at the University of Texas at Dallas. Values for the various constants stored in file IESCNTRLFILE are provided by them.]

2. H+ Mode.

At this time, only the first two offset voltages are processed. These values are normal mode horizontal and vertical (or vertical and horizontal) ion velocity measurements, which are converted from voltages to velocities in the same manner as specified above. The data reduction algorithms for the rest of the H+ mode data are TBD.

3. DM FIBA Mode.

The voltage from any of the filters is converted to an estimate of the power spectral density (PDS) by first converting it to an estimate of the RMS variance in the DM current for the filter frequency from:

$$Log_{10}\,(RMS(\Delta I))_j = G_j V_j - V_{j0}$$

where: $(RMS(\Delta I))_j$ is the RMS variance in the DM current at a frequency corresponding to the j^{th} filter (j = 1-NDMFLT),
G_j is the gain of the amplifier for the j^{th} filter,
V_j is the voltage output from the j^{th} filter,
V_{j0} is the voltage offset for the j^{th} filter,

This is converted to an estimate of the RMS variance in ΔN_i by:

$$(RMS(\Delta N_i))_j = \frac{(RMS(\Delta I))_j}{A_e\,(v_s + u_r\,)}$$

where: A_e is the effective area of the sensor,
v_s is the satellite velocity,
u_r is the ram ion drift velocity.

The RMS variance is then converted to an estimate of the PDS at the frequency of the j^{th} filter by:

$$PDS_j = \frac{(RMS(\Delta N_i))^2{}_j}{B_j}$$

where: B_j is the bandwidth (Hz) of the j^{th} filter.

This algorithm is implemented in Subroutine DMFIBA.

B.4 Scintillation Meter Processing Algorithms.

a. Sensor Description.

The ion Scintillation Meter (SM) is a planar Faraday cup sensor located on the ion-sensor ground plane with the ion RPA and ion Drift Meter sensors. The sensor is oriented with its normal parallel to the spacecraft direction-of-travel ($+y_s$). The output of the sensor collector is a current which is proportional to the ion density being swept into the collector, due primarily to the motion of the satellite. Volume 1, Technical Description, Figure 3.4.4 shows a cross section of the SM sensor. The current is converted to a voltage in an electrometer (EL), which can be set in any of five sensitivity ranges. The electrometer output is split into three paths, one to a difference amplifier, one to an analog multiplexer, and one to a wideband ranging amplifier. The difference amplifier (AMP) subtracts a base level voltage (from a previous EL output) from the latest EL output, amplifies it by (roughly) a factor of 10, and sends the amplified voltage to the analog multiplexer with the raw EL output. The wideband ranging amplifier can also be set to any of five sensitivity ranges. The amplified output is based to a bank, or comb, of band-pass filters/amplifiers which output voltages proportional to the variance in density at 9 (6, for IES2 and later configurations) specific fluctuating frequencies. (Since the satellite is moving, the frequencies can be converted to equivalent scale sizes by assuming that all variations measured are spatial and dividing the satellite velocity by the filter center frequencies.)

b. Sensor Output.

1. Density. The SM sensor provides 24 voltages per second from either the EL or AMP outputs, which can be converted to ion density. Imbedded in this stream are flags which indicate when the analog multiplexer has switched between the EL or AMP outputs and when the EL sensor has changed its range setting. The EL/AMP sequences used by the SM sensor and the various flags used to indicate status changes are quite complex. The basic description of these can be found in Volume 1, Technical Description, Section 6.3. The normal sequence, used when the ion density variations are small, starts at the first SM EL/AMP word with two zero flags indicating an AMP-to-EL source change (which may, on rare occasions, be a single zero flag), followed by two readings from the EL output, followed by a 5.11-volt flag indicating an EL-to-AMP source change, followed by AMP readings until 8 cycle 1 data frames have passed (15 or 16 seconds). At that time, the sequence repeats itself. If the AMP output overranges (i.e., the AMP output converts to a telemetry voltage less than 0.0 or greater than 5.11), two zero flags are reported followed by EL readings until the next cycle 1 data frame. At that time, a 5.11 flag is reported followed by AMP samples. If the AMP does not overrange in the next second, the sequencing continues until the next normal switch to EL is required or an overrange occurs. If the AMP overranges within the next second after a reset from EL, the output will remain on the EL for the next 8 cycle 1 data frames (15 or 16 seconds) at which time a 5.11 flag will be reported and the output will switch to AMP. These same two sequences will be followed if the output is currently from the AMP and the electrometer requires a range change, except that a range data word will be reported rather than zero flags at the changeover. If an EL range change occurs while the output is on EL, the sequence will continue unaltered. NOTE: Be sure that this sequencing as described in reference 1.2.3d is fully understood before attempting to modify the ELAMP subroutine. Table B1 lists the (nominal) minimum and maximum ion densities which can be reported by the SM sensor for five values of the ram drift velocity.

2. Density variance. One sample is output from each of the nine (six, for IES2 and subsequent configurations) filters every second.

Table B1 SM sensor density ranges

u_r	max N_i	min N_i
-2	$1.4x10^6$	$8.0x10^2$
-1	$1.2x10^6$	$6.7x10^2$
0	$1.0x10^6$	$5.8x10^2$
+1	$0.9x10^6$	$5.1x10^2$
+2	$0.8x10^6$	$4.6x10^2$
km/s	ion/cm^3	ion/cm^3

Note: These values assume a spacecraft velocity of 7.44 km/s and use the nominal EL amplifier sensitivity ranges listed in reference 1.2.3d.

3. Range data. A single range data word is output once every second as part of the filter comb output. This word indicates the range setting.

c. Data Extraction Algorithms.

1. Density data. As can be seen from the above description of the sensor, extraction and identification of the EL/AMP data is quite complicated. It is complicated by the expectation that an occasional data frame will be missing, which makes it possible to miss a change between EL and AMP. An additional complication is that the range and status change flags imbedded in the EL/AMP data stream are not necessarily uniquely different from EL/AMP voltages, although this problem will occur only in well defined special cases (at the high density end of the least sensitive EL range setting and the low density end of the most sensitive EL range setting). The steps followed in extracting the data are covered in detail in the description of the ELAMP subroutine (section 2.4.4.1.3). The discussion here will focus on various quirks and oddities in the processing. (See note at end of this section)

(a) AMP zero-current bias. The AMP output is not an absolute measure of the current collected by the SM sensor. It is the difference between the latest output and the current output at the time the source of the EL/AMP output switched from EL to AMP. In order to calculate the total current, and thus the total ion density, the EL output current (or the density corresponding to this current) at the time of the switch must be available. This value must be retained from one data frame to the next until a new EL reading is obtained. In Subroutine ELAMP, this is saved in the form of a base ion density in variable DEN0. No AMP data will be processed until a DEN0 value is obtained.

(b) Amplifier range settings. The electrometer amplifier and the wideband ranging amplifier both have five sensitivity settings which are used to keep the reported voltages from the EL/AMP data and from the filter data within the range allowed by the 9-bit telemetry words (000 to 511). The EL amplifier (not to be confused with the differencing amplifier, denoted AMP) will automatically adjust the range setting to provide the most sensitive setting which will allow the voltages from measured current to remain within the telemetry band. The wideband (WB) amplifier can either operate in this mode or remain fixed at a single setting as commanded from the ground. The settings for both ranging amplifiers are reported once per data frame in the range data word (data frame word 16

in the Common Format) via a voltage which is converted to the range settings (1-5) using a look-up table. This look-up table is stored in file IESCNTRLFILE and is used in Subroutine RANGE to determine the range settings for the two amplifiers. The EL range can also be reported whenever it changes in the EL/AMP data stream. In this case the range setting is reported as one of five voltages nominally outside the allowed voltage range of EL/AMP density measurements. This voltage is converted to a range setting by a second look-up table (with only five entries) also stored in IESCNTRLFILE. NOTE: When either amplifier undergoes a range setting change, the filter data should not be used for that data frame.

(c) Missing data frames. If one or more data frames are skipped, there is some uncertainty as to the source of the EL/AMP data, since the source may have changed during the time covered by the missing frames, and uncertainty in the value to use as the zero-current bias for AMP data, as this also may have changed. Processing of the EL/AMP data is suspended until an unambiguous status change flag is found in all cases save one - if the data source was from EL in the last good data frame, no range changes have occurred (as indicated in the range data word) since the last good data frame, and the EL sequence counter (variable NEL) indicates that the data source should still be EL. **Then, and only then,** will processing of the EL/AMP data continue with the first good data frame after the missing data frames without requiring a status change flag.

(d) EL range-flag/data ambiguities. The imbedded status flags in the EL/AMP data stream are nominally 5.11 volts for an EL-to-AMP switch, 0.0 volts for an AMP-to-EL change with no coincident EL range change, and a range of voltages from 0.12 to 0.28 volts for a EL range change. The range of voltages from the EL output corresponding to density measurements is constrained to 0.3-5.0 volts with two exceptions: 1) when the EL amplifier range setting is at its least sensitive (range 5), the output voltage is no longer constrained to remain below 5.0 volts (high-density ambiguity), and 2) when the EL amplifier range setting is at its most sensitive (range 1), the output voltage is no longer constrained to remain above 0.3 volts (low-density ambiguity). In the high-density case, an EL-to-AMP flag should only occur in the first five EL/AMP data words in a data frame. Therefore, an apparent EL-to-AMP flag located elsewhere when the EL range is 5 is processed as density data. If it is in the first five words, the processing is switched to AMP and a flag is set to check the next data point to make sure that this was not just a high-density measurement. In the low-density case, if the data source is currently EL and the range data word shows no change in the range setting during the current data frame, any voltages below 0.3 are assumed to be density measurements. If the output source is AMP, any voltage below 0.3 is assumed to be an AMP-to-EL change with a possible EL range change.

2. Density variance data. Extraction and interpretation of the filter data is much less complicated. The only special case which needs to be accounted for in extracting the data is one mentioned earlier: the filter data should not be used from any data frame in which either the EL or WB ranging amplifiers change range.

d. Data Reduction Algorithms.

1. Density. The voltage from EL or AMP is converted to ion density by one of the following equations:

$$N_i = \frac{G_{EL} \ x \ V_{EL}}{A_e \ (v_s + u_r \ )} \qquad (EL\ data)$$

or

$$N_i = N_{i0} + \frac{G_{EL} \ x \ V_{AMP}}{A_e(v_s + u_r \ )} \qquad (AMP\ data)$$

where: V_{EL}, V_{AMP} are voltages from the EL or AMP output, respectively,
v_s is the satellite velocity,
u_r is the ion ram drift velocity,
A_e is the effective area of the sensor,
N_{i0} is the density calculated from the last EL density measurement,
G_{EL} is a gain factor for the electrometer, given by:

$$G_{EL} = G_0 \ x \ \sqrt{10}^{\ (J_{EL} - 1)}$$

where: G_0 is the gain factor for the most sensitive range setting,
J_{EL} is the range setting number (1-5, where $J_{EL} = 1$ is the most sensitive setting).

The algorithms to calculate density and gain are implemented in Subroutine ELAMP.

2. Density variance. The voltage from any of the filters is converted to an estimate of the power spectral density (PDS) by first converting it to an estimate of the RMS variance in the EL current for the filter frequency from:

$$Log_{10} \ (RMS(\Delta I))_j = G_j V_j - V_{j0} - \frac{(2 - J_{EL} \ J_{WB} \ )}{2}$$

where: $(RMS(\Delta I))_j$ is the RMS variance in the EL current at a frequency corresponding to the j^{th} filter (j = 1-NSMFLT),
G_j is the gain of the amplifier for the j^{th} filter,
V_j is the voltage output from the j^{th} filter,
V_{j0} is the voltage offset for the j^{th} filter,
J_{EL} is the range setting number for the electrometer amplifier,
J_{WB} is the range setting number for the wideband ranging amplifier (1-5, where $J_{WB} = 1$ is the most sensitive setting).

This is converted to an estimate of the RMS variance in ΔN_i by:

$$(RMS(\Delta N_i))_j = \frac{(RMS(\Delta I))_j}{A_e \ (v_s + u_r \ )}$$

which is then converted to an estimate of the PDS at the frequency of the j^{th} filter by:

$$PDS_j = \frac{(RMS(\Delta N_i))^2_{\ j}}{B_j}$$

where: B_j is the bandwidth (Hz) of the j^{th} filter.

This algorithm is implemented in Subroutine FILTER.

NOTE:

The preceding discussions of processing the SM EL/AMP data are based on the designed behavior of the various range flags. During the early orbit checkout of the SSIES package on satellite F8 it became apparent that both the imbedded flags and the range data flag were "floating" away from their nominal voltage values. In particular, the 0.0-volt flag which was to indicate a change from the AMP to the EL with no associated range change was being reported anywhere in the range from 0.01 to 0.12 volts, which overlaps the lowest voltage of the imbedded range change flags. These flags, in turn, were floating roughly 0.02 to 0.04 volts above their nominal values, and the range data flag was floating roughly 0.03 to 0.06 volts above its nominal values. The ELAMP and RANGE routines were completely rewritten to allow processing the data as it was being reported, but at a cost of being unable to process all of the data. There will be times when the software cannot differentiate between a floating flag-voltage and a density-data voltage. At these times, the processor will place itself into a wait status until a totally unambiguous 5.11 volts EL-to-AMP flag is found in either data word 1 or data word 5 (see function FND511).

B.5 C_kL Calculation Algorithm.

IMPORTANT NOTE: In the following discussion, the assumption is made that the ionospheric plasma is locally neutral, i.e., the electron density equals the ion density equals the plasma density.

a. Background.

Small-scale irregularities in the ionospheric plasma density can be characterized in their behavior in the time (or space) domain by parameters which describe the variance of the observed density from a background trend, and in the frequency (or wave-number) domain by parameters which describe the power density spectrum (PDS) of the irregularities. The usual parameters measured in the time domain are the RMS (root mean square) variation of the density about the trend, RMS(ΔN), and the relative RMS variation, (RMS(ΔN))/N, where N is the average density for the data sample. Figure B3 shows a fairly typical (one-dimensional) PDS for ionospheric irregularities which shows the power-law nature of the PDS, i.e., the PDS can be adequately modeled by a power-law distribution of the form:

$$PDS(f) = T_1 f^{-p_1}$$

On a log-log plot, such as Figure B3, this will be a straight line with a slope $-p_1$ and a value of T_1 at a frequency of 1 Hz. These two parameters are those typically used to characterize the frequency domain behavior of the irregularities.

Both of these descriptions of the irregularities are not wholly adequate to describe the irregularities for three reasons. First, the measurements from which they are taken are necessarily one-dimensional (a probe moving along a single axis through a three-dimensional medium), while the irregularities are three-dimensional in distribution and shape. Second, both descriptions have folded into them factors involving the speed of the probe with respect to the irregularities in a coordinate system defined by the orientation and shape of the irregularities. Thus, a probe travelling through an irregularity patch in one direction, such as along the local geomagnetic field direction, will provide quite different measures for RMS(ΔN) and (T_1, p_1) than the same probe moving through the same irregularity patch across the local field direction. Finally, the RMS(ΔN) and (RMS(ΔN))/N parameters will be a function of the size of the data sample from which the measurement is taken and the characteristics of the calculation and removal of the background trend. This will also affect the measurements of T_1 and p_1, but not as severely.

Either RMS(ΔN) or the (T_1, p_1) set can be used to calculate a fifth, and final, parameter which describes the three dimensional power law distribution of the irregularities, C_k. This parameter is designed as the power spectral density, or irregularity "strength," of the three dimensional irregularity PDS at a scale size of one kilometer. (Note: This parameter is identical in concept to the traditional C_k parameter which is defined at a scale size of 2 meters.) It can be calculated, in theory, from either the RMS(ΔN) or (T_1, p_1) observations via the following equations:

$$C_k = 5 \; x \; 10^8 \; \Gamma\left(-\frac{3}{2}\right) p_1 \left(\frac{10^3}{\alpha}\right)^{p_1 - 1} (RMS(\Delta N))^2 \qquad \textbf{(B-01)}$$

$$C_k = 5 \; x \; 10^8 \; p_1 \left(\frac{10^3}{V_{eff}}\right)^{p_1 - 1} T_1 \qquad \textbf{(B-02)}$$

where: α is the largest scale size (known as the outer scale size),

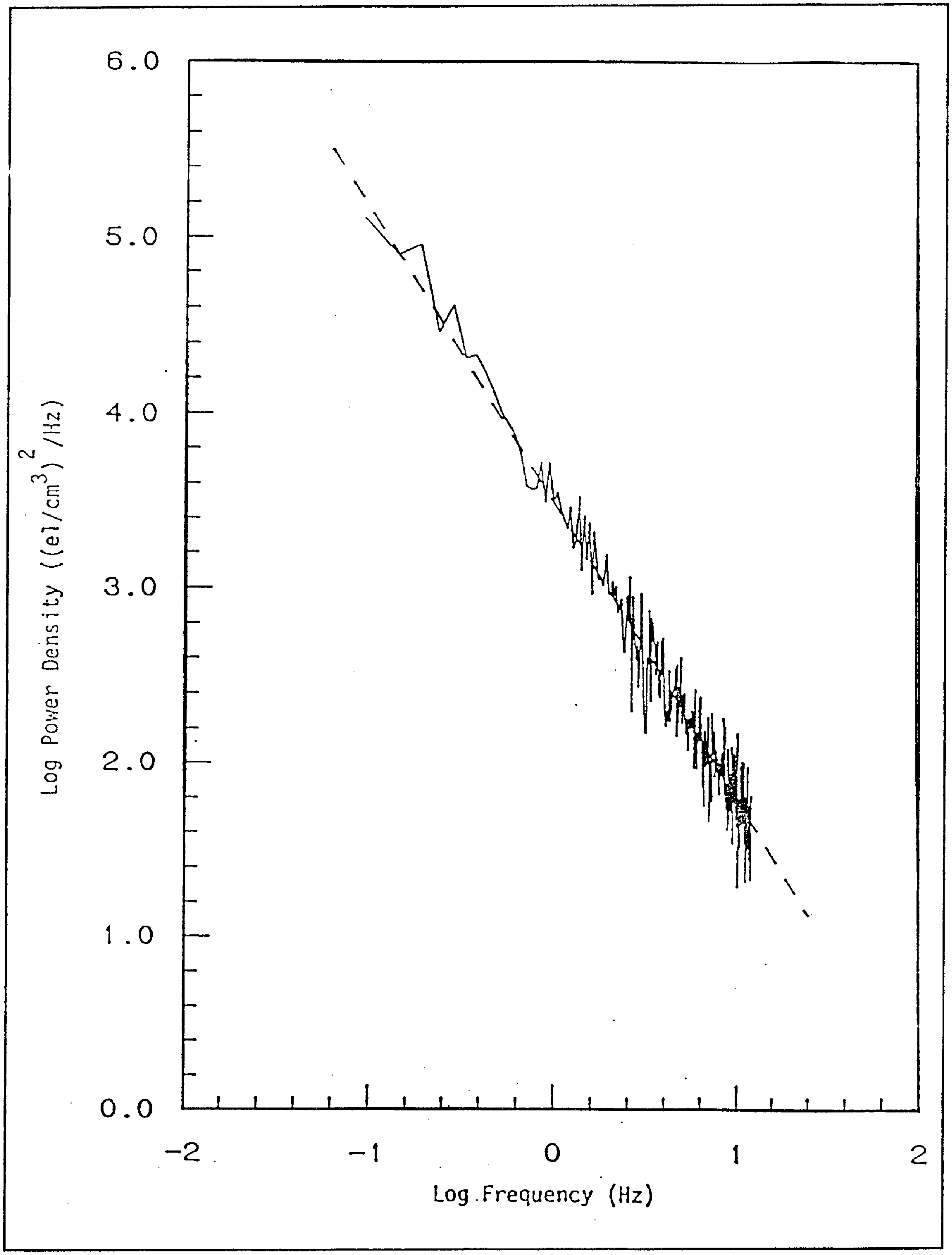


Figure B3 Sample irregularity power density spectrum

Γ	is the gamma function,
V_{eff}	is the velocity of the probe with respect to the irregularities in the irregularity coordinate system.

While C_k can be calculated from the RMS(ΔN) using Equation B-01, selecting a proper value for the outer scale size provides an additional level of uncertainty in the calculation. In practice, α is usually specified by the size of the data set from which the RMS(ΔN) was measured or by the largest scale size remaining after the trend is removed. In the APGA program, C_k is calculated from measurements of T_1 and p_1 using Equation B-02. This avoids the outer scale question and, if the data preparation is done carefully, can provide reasonably good estimates for C_k from finite data samples.

In order to use the C_k parameter for estimating the effects of the irregularities on transionospheric radiowave propagation, it must be converted to a path integrated strength, commonly denoted C_kL (C_sL). Calculating C_kL from C_k measured at the DMSP altitude is similar in concept, although practically more difficult, to calculating an estimate for Total Electron Content (TEC) from a single density measurement made at the same altitude. If the measurement is made near the F2 peak, C_kL can be reasonably estimated by multiplying C_k by the ionospheric slab thickness (TEC divided by the F2 peak density). At higher altitudes, and definitely at DMSP altitudes, the conversion from C_k to C_kL must account not only for the thickness of the irregularity layer, but also for the variation of C_k with height. Neither of these is well known, so the APGA program uses a very simple multiplicative factor, an effective thickness, L_{eff}, to account for both effects. This factor is implemented such that it can have three different values at equatorial, middle, and high latitudes with smooth transitions between latitudinal regimes. Initial values for L_{eff} were calculated for each regime for typical nighttime profiles assuming that:

a. the topside density profile followed a Chapman distribution,
b. the irregularity layer reached from the F2 peak to (at least) the satellite altitude,
c. the parameter (RMS(ΔN))/N is constant throughout the irregularity layer.

Values for L_{eff} were 1.4×10^7, 4.5×10^6, and 3.5×10^6 for the equatorial, middle, and high altitude regimes, respectively. This conversion from C_k to C_kL is definitely the weak link in the calculation of C_kL. A great deal of further work is required to provide better values for L_{eff}.

(NOTE: The preceding description of the characterization of ionospheric irregularities was, of necessity, brief. A full description, based on the C_kL parameter but still applicable, can be found in references 1.2.3s and 1.2.3v, and in other references found in the C_kL section of the SSIES Project Workbook.)

b. Calculation Procedure.

(1) Collect ion density data and filter PDS data from the SM sensor or electron density data from the EP sensor (modes C, D, and DS only) until 22 seconds of data have been collected.

(2) Check the 512-point data set centered in the 22 seconds for missing data points. Fill in any single or double missing data points by linear interpolation from the points on either side of the missing data. If more than two sequential missing data points are found in the 512-point set, fall back to the 256-point set centered in the 22 seconds. If more than two sequential missing data points are found in this set, or if more than 25 total missing data points were found in either the 256 or 512 point sets, do not attempt to calculate irregularity parameters from the data set. Otherwise, move on to the next step with either a 512 or 256 point data set for further processing.

(3) Calculate the density trend from a quadratic least-squares fit to the density data. Subtract the trend from the data to provide the detrended data sample, ΔN.

(4) Calculate RMS(ΔN) and (RMS(ΔN))/N from the center ten seconds of the detrended data set (240 points).

(5) If the (RMS(ΔN))/N is below the user selected threshold for the C_kL calculation, quit processing the data at this point.

(6) Apply a "split-bell" cosine taper window to the detrended data set. The severity of the taper (in percent of the data effected by the window) is specified by the user in the range 0% (no window) to 100%.

(7) Calculate the PDS of the data set from a Fast Fourier Transform (FFT) of the detrended, windowed data set. This will cover the frequency range 0.047 - 11.95 Hz.

(8) If user selected, smooth the PDS using a moving, centered smoother with binomial weights.

(9) Calculate a decimated PDS by resampling the full PDS at constant log-frequency intervals ($\Delta \log f = 0.334$).

(10) Add the PDS estimates from the SM sensor filter comb for the 10-second period in the center of the data set to the FFT-produced (full) PDS, and the average value for each filter to the decimated PDS. (Note: The log-frequency steps between the peaks in the response functions of the filters in the filter comb are identical to the constant log-frequency step used in re-sampling the full FFT-produced PDS).

(11) Calculate T_1 and p_1 from a log-linear fit to either the full or decimated PDS (user selected) over a frequency range specified by the user. If the T_1 parameter is out of range, do not calculate C_kL. If the p_1 parameter is out of range, either use a model value for p_1 or do not calculate C_kL as specified by the user.

(12) Calculate the effective velocity of the probe in the irregularity coordinate system using the spacecraft velocity and the velocity of the ions with respect to the spacecraft as measured by the DM and RPA sensors, and the orientation and shape (axial ratios) of the irregularities from the WBMOD irregularity model.

(13) Calculate C_k from Equation B-02.

(14) Convert C_k to C_kL by multiplying C_k by a model value for the effective thickness, L_{eff}.

B.6 Microprocessor Sweep Analyses Processing.

a. Description.

The SSIES instrument package is controlled by a microprocessor which is located in the SSIES main electronics package. This microprocessor also analyzes the EP and RPA sweeps to determine electron and ion densities and temperatures, the potential of the EP and RPA sensors, and the ion ram drift velocity. It is not precisely known what algorithms are used for these analyses, but they should be comparable to those documented in references 1.2.3b and 1.2.3c. The EP sweep analyses are run in the first second following completion of the sweep, and the RPA sweep analyses are run in the second second following completion of the sweep.

b. Output.

The results reported from the EP sweep analyses are the electron temperature (data frame word 6, cycle 1 in the Common Format), electron density (word 6, cycle 2), the EP sensor potential (word 13, cycle 1), and a set of 9 sweep processing flags (packed in word 11, cycle 1). The results for an EP sweep analysis are first reported in the first cycle 1 frame after the end of the sweep. If the EP sensor is in a 4 second sweep mode (modes A, B, BS, C, D, or DS) or in a calibration sweep, the analysis results will appear twice. If the EP sensor is in mode E, the analysis results will only appear once.

The results reported from the RPA sweep analyses are the O+ ion density (word 5, cycle 2 of the Common Format), the light ion density (word 4, cycle 2), the O+ ion temperature (word 5, cycle 1), the light ion temperature (word 4, cycle 1), the ion ram drift velocity (word 14), the RPA sensor potential (word 13, cycle 2), and a set of 9 sweep processing flags (packed in word 11, cycle 2). The results for an RPA sweep analysis are first reported in the first cycle 2 frame after the end of a sweep. Each analysis result will appear twice.

Although the sweep analyses are nominally available when indicated, they may be one data frame late if the analysis takes longer than the allotted time. In this case, the analysis from the previous sweep will continue to be reported until a new sweep result is available. This will be more of a problem with the RPA analyses than with the EP analyses. In addition, when the EP sensor is in a DC mode, the last EP sweep analysis made will be reported throughout the time when the sensor is in non-sweep operation.

c. Data Extraction Algorithms.

(1) EP Sweeps. Extraction of the EP sweep analyses is greatly exacerbated by the different sweep periods available (4 seconds for normal sweeps in modes A, B, BS, C, D, and DS; 2 seconds for normal sweeps in mode E; 3 seconds for calibration sweeps in all modes; 0 seconds (no sweep) for most of modes C, D, and DS) and the fact that the different sweep times are mixed throughout the modes (i.e., modes A and B can have 4 and 3 second sweeps; modes C and D can have 4, 30, and 0 second sweeps; mode E can have 2 and 3 second sweeps). In order to efficiently extract the sweep information, the processor must keep track of when to look for a sweep analysis and how long the sweep analysis will be available. This is done by following the value of the sweep clock counter and watching for the introduction of calibration sweeps via the setting of bit 6 (counting down from the most significant bit in the 9-bit word) in data frame word 1, cycle 1 of the Common Format. In one of the DC modes (modes C, D, and DS), an additional complication arises in that the processor must know not to extract analyses when no sweeps have occurred in order to avoid repeating a sweep analysis. This is done via three logical variables, SWPNOW which is .TRUE. if the EP sensor was sweeping during the

current data frame, NOSWP which is .TRUE. when there is no sweep analysis to be extracted, and BIAS which is .TRUE. when the current EP sweep is in calibration mode.

(2) RPA Sweeps. Since the RPA sensor sweeps in a constant, never varying 4 seconds per sweep period, the extraction is much simpler than for the EP sweeps. Collection of sweep analysis data begins for a particular sweep in the first cycle 2 data frame of each 4 second period as specified by the sweep clock cycle counter. The value of the potential of the ion sensor ground plane (V_{bias}) must be saved from the previous 4 second period to convert from sensor potential to spacecraft potential. When the data from both cycle 1 and cycle 2 have been collected, the processor waits until the first cycle 1 frame at the start of each 4 second period when it indicates that an analysis has been received.

d. Data Reduction Algorithms.

Other than converting from data frame words to engineering data, the following data reduction algorithms are used:

(1) Ion densities:

$$N_i = 10.0^{(\log N_i)} \qquad ion/cm^3$$

where: log N_i is the log density value reported in the data frame word.

(2) Electron density:

$$N_e = \frac{3.476 \times 10^{13}}{T_e} \; 10.0^{(\log N_e)} \qquad el/cm^3$$

where: T_e is the electron temperature (c. 1000° K if the temperature analysis was not reported),
log N_e is the log density parameter reported in the data frame word.